

# СОДЕРЖАНИЕ

Введение.....	5
1 Анализ предметной области.....	5
2 Постановка задачи проектирования.....	12
2.1 Принятие допущений .....	12
2.2 Постановка задачи 1 .....	13
2.3 Постановка задачи 2 .....	13
3 Описание структурной схемы САПР .....	13
4 Описание схемы работы САПР .....	18
5 Описание видов обеспечений .....	19
5.1 Математическое обеспечение.....	19
5.2 Информационное обеспечение.....	43
5.3 Лингвистическое обеспечение.....	45
5.4 Программное обеспечение.....	46
5.5 Техническое обеспечение .....	47
6 Результат работы САПР.....	48
6.1 Демонстрация результата.....	48
6.2 Анализ полученного результата.....	54
Заключение .....	55
Список используемых источников.....	55
Приложение А Структурная схема САПР.....	56
Приложение Б Схема работы САПР .....	57
Приложение В Даталогическая модель САПР.....	58
Приложение Г Схема алгоритма.....	59
Приложение Д Результат работы САПР.....	60

ТГТУ.230100.Б01 ПЗ

Изм. Лист № докум. Подп. Дат.

Разраб. Шмапов

Касьянов

Реценз.

Дьяков

Утверд.

Митованов

Пояснительная  
записка

Лист Листов

1 58

САПР, гр. БСП-41

## ВВЕДЕНИЕ

Необходимой частью коммерческой деятельности любого предприятия является система товароснабжения, которая включает экономические, организационно–правовые отношения между поставщиком и потребителями. В свою очередь, важнейшим элементом системы товароснабжения является транспортировка товаров. Возникает потребность в организации процесса перевозок.

В коммерческой деятельности от транспорта во многом зависит не только скорость доставки товара потребителям, но их сохранность, качество, а также затраты по перевозке, которые занимают большой удельный вес в издержке обращения. Высокая оперативность, надежность и стабильность доставки товаров из пунктов производства в пункты распределения и потребления с минимальными затратами, а также без потерь количества и качества этих товаров является первостепенным условием эффективной коммерческой деятельности /1/.

## 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

«Бастион-Продукт» - это оптовая база, осуществляющая торговую деятельность. В результате этой деятельности производится:

- закупка продовольственных товаров
- продажа продовольственных товаров
- предоставление услуг доставки товара

Товаром являются продукты питания: алкоголь, шоколад, маргарин, крупа, макаронные изделия, подсолнечное масло, майонез. Перечень товаров соответствует требованиям рынка. Спрос на некоторые виды товаров обладает свойством сезонности.

Весь товар хранится на складах. В распоряжении базы имеется 6 складов. Все склады располагаются на территории базы в городе Тамбове. Каждый склад приспособлен для конкретного вида продуктов, исходя из условий хранения.

Например, для хранения маргарина, майонеза, шоколада необходимы соответствующие температурные условия. Кроме того, деление складов по видам продукции обеспечивает удобство учета товара и обслуживания клиентов.

«Бастион-Продукт» имеет более 1500 постоянных клиентов. Клиентами являются частные либо индивидуальные предприниматели.

Процесс взаимодействия клиента с базой можно представлен на рисунке 1.



Рисунок 1.1 – Схема взаимодействия между базой и клиентом.

Существует несколько способов организации заказа клиентом:

- 1) непосредственно на базе. В этом случае необходимо личное присутствие клиента;
- 2) при помощи телефона либо факса;
- 3) осуществление заказа при помощи экспедитора базы.

После поступления заказа производится его обработка отделом выписки товара. Если указанного клиентом товара нет в наличии, то заказ не будет осуществлен, а соответственно информация о заказе не зафиксировывается в базе данных заказов.

Очень важным фактом является, что объемы заказов отличаются в каждый период времени. Это связано с тем, что клиент осуществляет заказ, если наблюдается необходимость в каком-либо виде товаров. Данные о заказах принимаются и обрабатываются в течение рабочего дня (с 9.00 до 17.00). В результате обработки заказов формируются накладные. Накладная содержит информацию о товаре (наименование, количество, стоимость) и о заказчике (юридический адрес, адрес доставки, имя клиента, реквизиты). По объемам заказов клиенты подразделяются на оптовых и розничных.

Оплата товара производится может производиться несколькими способами:

- 1) безналичный расчет;
- 2) расчет наличными средствами на базе;
- 3) расчет наличными средствами с экспедитором базы.

Также оплата может производиться как предварительно, так и по факту доставки товара.

В случае, когда клиент не способен осуществить доставку самостоятельно, база предоставляет услугу доставки товара.

Доставка осуществляется грузовыми автомобилями. «Бастион-Продукт» располагает собственным автопарком, насчитывающим 26 автомобилей. 3 автомобиля марки ЗИЛ грузоподъемностью 5 тонн отданы на нужды отдела, занимающегося оптовыми клиентами. Таким образом, оставшиеся автомобили, в количестве 23 штук, отданы на нужды отдела, занимающегося розничными клиентами.

Структура автомобильного парка:

- 2 автомобиля марки ИФА грузоподъемностью 6 тонн;
- 2 автомобиля марки ЗИЛ грузоподъемностью 5 тонн;
- 1 автомобиль марки ГАЗ грузоподъемностью 3.5 тонны;
- 18 автомобилей марки ГАЗ грузоподъемностью 1.5 тонны.

В экипаж грузового автомобиля, как правило, входят 2 человека: водитель и экспедитор. Водитель осуществляет управление транспортным средством. К обязанностям экспедитора относится набор товара для каждого заказа на складе и транспортировка сопроводительных документов. Кроме того экспедитор осуществляет прием заказов и наличных денежных средств за доставленный товар (в случае, если оплата осуществляется по факту доставки) от клиентов. В настоящий момент штат базы располагает 16 экспедиторами.

Водитель и экспедитор имеют фиксированный 8-часовой рабочий день. За каждый час работы сверхурочно существует повышенный тариф оплаты.

Для осуществления доставки необходимо организовать загрузку каждого автомобиля и разработать маршрутный лист для каждого автомобиля.

Клиенты располагаются на территории всей Тамбовской области, как в городах(например Тамбов, Моршанск, Мичуринск), так и в селах и деревнях. Плотность расположения клиентов в области неоднородна. Очень велика плотность расположения клиентов в городах. Например, в Моршанском районе большинство клиентов располагается в г.Моршанске(около 200 клиентов), а в Пичаевском районе всего около 40 клиентов, которые располагаются в 25 населенных пунктах. Каждый клиент имеет юридический адрес и адрес, по которому осуществляется доставка.

Задача организации грузовых перевозок (доставки товара) можно разделить на 2 подзадачи:

- 1) определения перечня клиентов, обслуживаемых грузовым автомобилем, из расчета массы и объема товара;
- 2) определение оптимального маршрута для каждого автомобиля.

В результате решения задачи формируется маршрутный лист. Затем необходимо произвести загрузку каждого автомобиля. Загрузка товаров в кузов автомобиля производится в порядке, обратном порядку посещения клиента, указанном в маршрутном листе. Порядок загрузки товаров для каждого клиента определяется видом товара и условиями его перевозки.

В настоящее время организацией грузовых перевозок занимается диспетчер, находящийся на базе. Причем, диспетчер принимает решение лишь на счет перечня клиентов и выбирает грузовой автомобиль для обслуживания этих клиентов. Вся область была поделена диспетчером на районы доставки. Основной задачей данного деления является организация клиентов в группы по следующим признакам:

- 1) географическое положение;
- 2) объемы заказов клиентов;
- 3) количество клиентов.

Таким образом, область была поделена на 22 района доставки. Крайне важным является то, что при уже сформированных районах доставки перечень клиентов всегда является разным (это связано с потребностями каждого клиента).

Порядок же посещения клиентов определяется водителем непосредственно перед отправлением в рейс.

Доставка товара осуществляется по существующему расписанию 5 дней в неделю. Этого расписания придерживается база и оно устраивает клиентов. Также расписание является фиксированным и редко претерпевает изменения. Таким образом, независимо от того, когда был осуществлен заказ, доставка товара будет произведена по расписанию.

На данный момент база не обладает автоматизированной системой организации грузовых перевозок, а применяемые ранее системы продемонстрировали свою несостоятельность. Таким образом, все решения по поводу транспортировки товаров принимаются людьми. При этом чаще всего возникают трудности, связанные с большой трудоемкостью при расчетах маршрутов. Кроме того найденные решения не всегда являются оптимальными.

Целью данной работы является:

- исследование методов оптимизации грузовых перевозок;
- разработка автоматизированной системы расчета оптимального маршрута.

При расчете оптимального маршрута существует несколько критериев.

1) Время максимальной загрузки автомобиля в процессе доставки заказов. Очевидно, что сначала необходимо посетить того клиента, от которого пришел наибольший с точки зрения массы товара заказ. Таким образом, удастся уменьшить время максимальной загрузки автомобиля. Это позволит уменьшить износ деталей машины и увеличить среднюю скорость движения.

2) Время работы водителя.

3) Качество дороги. При составлении маршрута передвижения нельзя исключать такого понятия, как качество дороги, по которой оно осуществляется. Очевидно, что можно пожертвовать протяженностью пути в пользу качества дороги, в следствие того, что движение по плохим дорогам может привести к выходу из строя отдельных частей машины и их повышенному износу. В результате может оказаться, что сумма средств, затраченных на восстановление

машины, больше, чем сумма сэкомленных средств. Кроме того велика вероятность порчи товара во время перевозки из-за недостатков качества дороги.

4) Время прибытия машины в пункт назначения. В следствие того, что большинство клиентов располагается в деревнях и небольших селах, актуальным становится вопрос о времени доставки товара, так как большинство клиентов имеет индивидуальных график работы и маршрут доставки заказов следует проектировать с учетом времени пребывания в конкретный пункт назначения. Это позволит избежать повторного посещения данного клиента в другой день и сохранить репутацию фирмы.

5) Длина пути. Одним из важнейших критериев организации доставки заказов является величина пути, затраченного на посещение всех клиентов. Очевидно, что чем короче общий путь передвижения, тем меньше средств будет затрачено на топливо для машины.

Кроме того очень важным является оборудование кузова автомобиля. Например, при высоких температурах окружающей среды для перевозки шоколада необходимо соответствующее оборудование для поддержания более низкой температуры перевозимого товара. «Бастион-продукт» не обладает подобным оборудованием. Таким образом, например, невозможна транспортировка шоколада летом при высоких температурах, а также пива зимой при низких температурах. В том случае, если существует острая необходимость в транспортировке скоропортящихся видов продукции, отдел доставки прибегает к услугам сторонних водителей, обладающих соответствующим оборудованием. Однако такое применяется в редких случаях и чаще всего при работе с оптовыми клиентами.

При расчете оптимального маршрута можно учитывать как все критерии, так и отдельные из них.

В данной работе критерием расчета оптимального маршрута является длина пути, пройденного автомобилем в результате посещения всех клиентов. Основная задача разработанной подсистемы – это расчет такого маршрута передвижения, при котором путь, пройденный в результате посещения всех клиентов по одному разу, будет минимален.

Далее будем говорить не о положении клиента, а о положении населенного пункта, в котором располагается клиент. Если в одном населенном пункте находится несколько клиентов, то очевидно, что расстояние между клиентами внутри населенного пункта мало по сравнению с расстоянием между двумя населенными пунктами. Также очевидно, что необходимо произвести сначала посещение всех клиентов внутри одного населенного пункта, прежде чем переезжать в другой.

Положение каждого населенного пункта характеризуется 2-мя координатами: координатой широты и долготы /5/. Зная, что одному градусу широты соответствует расстояние в 111.12 км, а одному градусу долготы соответствует расстояние  $111.12 \cdot \cos(\text{значение широты})$  км, можно найти расстояние между всеми клиентами. Однако это расстояние нельзя считать фактическим расстоянием, пройденным автомобилем для перемещения из одного пункта в другой. Это связано с тем, что движение может осуществляться только по дорогам. Их расположение не меняется с течением времени. Следовательно, теперь для определения расстояния между двумя пунктами необходимо знать длину дороги, соединяющей их. В силу наличия нескольких вариантов проезда между 2-мя населенными пунктами, необходимо определить какой именно из них будет рассматриваться. Поскольку в данной работе во внимание принимается лишь длина пути, то будем считать, что расстоянием между двумя пунктами будет являться кратчайший вариант проезда между этими пунктами. Вариантом проезда в данном случае является набор участков дорог, последовательное посещение которых приведет к перемещению из одного пункта в другой.

Таким образом, задача организации оптимального маршрута можно разделить на 2 подзадачи:

- 1) определение кратчайших вариантов проезда между населенными пунктами;
- 2) определение порядка посещения населенных пунктов.



## 2 ПОСТАНОВКА ЗАДАЧИ ПРОЕКТИРОВАНИЯ

### 2.1 Принятие допущений

Введем определение следующих понятий:

1) пункт назначения – это населенный пункт, в котором производится отгрузка товара;

2) путевая точка – это точка на карте, обозначающая поворот дороги или перекресток;

3) отрезок дороги – это отрезок прямой на карте, концами которого являются 2 путевые точки, путевая точка и пункт назначения, либо 2 пункта назначения.

Пусть  $A$  – множество пунктов назначения, а  $B$  – множество путевых точек. Введем такое множество  $C$ , что:

$$A \in C$$

$$B \in C$$

Тогда подмножество  $v$ , состоящее из 2 элементов  $a$  и  $b$ , можно считать отрезком дороги, соединяющим  $a$  и  $b$ :

$$v = \{a, b\}$$

$$a, b \in C$$

Пусть  $D$  – множество, соержащее все отрезки дорог.

$$v_i \in D$$

где  $i$  – целое число ( $i=1,2,\dots,m$ ).

Вариантом проезда  $S_{ij}^r$  ( $r$ -целое число;  $r>0$ ) между пунктами назначения  $a_i$  и  $a_j$  называется множество, состоящее из подмножеств  $l_k$  ( $k$ -целое число;  $k>0$ ), причем выполняется условие:

$$S_{ij}^r \in S_{ij} \in D$$

где  $S_{ij}$  – множество вариантов проезда между пунктами назначения  $a_i$  и  $a_j$ ;

$S_{ij}^r$  –  $r$ -й вариант проезда между пунктами назначения  $a_i$  и  $a_j$ .

Для всех  $v_k \in S_{ij}^r$  выполняются условия:

1) если  $v_k = \{x, y\}$ , то  $l_{k+1} = \{y, z\}$  при  $x, y, z \in B$ ;

2)  $v_1 = \{a_i, b\}$ ,  $a_i \in A, b \in B$ ;

3)  $v_n = \{b, a_j\}$ ,  $a_j \in A, b \in B$ .

$L(S_{ij}^r)$  – это длина  $r$ -го варианта проезда между пунктами назначения  $a_i$  и  $a_j$

## 2.2 Постановка задачи 1

Для каждой пары  $a_i, a_j \in A$  найти такое подмножество  $S_{ij}^r \in S_{ij}$ , чтобы

$$L(S_{ij}^r) \rightarrow \min$$

где  $S_{ij}^r \in S_{ij}$  –  $r$ -й вариант проезда между пунктами назначения  $a_i$  и  $a_j$ ;

$i, j$  – целые числа ( $i \neq j$ ;  $i, j = 1, 2, \dots, n$ ).

## 2.3 Постановка задачи 2

Пусть  $t$  – порядок посещения пунктов назначения  $a_i \in A$ , задающийся циклической перестановкой

$$t = (j_1, j_2, \dots, j_n, j_1)$$

где  $j_1, \dots, j_n$  – разные номера.

Расстояния  $L(S_{ij}^r)$  между парами пунктов  $a_i$  и  $a_j$  образуют матрицу  $E$ .

Найти такой порядок посещения пунктов  $t$ , который минимизирует функционал:

$$L(t) = \sum_{k=1}^n E_{j_k j_{k+1}}$$

При ограничениях:

1)  $E_{ij} > 0$ ;

2)  $E_{ii} = \infty$ ;

3)  $E_{ij} = E_{ji}$ ;

4)  $E_{ij} + E_{jk} \geq E_{ik}$ ;

## 3 ОПИСАНИЕ СТРУКТУРНОЙ СХЕМЫ САПР

В ходе анализа предметной области был выявлен необходимый перечень средств и подсистем необходимый для разработки автоматизированной системы планирования грузовых перевозок.

Таким образом, в ходе решения задачи разработки структурной схемы, было выявлено, что проектируемая САПР состоит из пяти основных подсистем.

1) Подсистема обработки исходных данных. Данная подсистема преобразует исходные данные, поступающие из базы данных в формат, требуемый программой. База данных представляет собой файл с расширением .xlsx. Данные в нем имеют следующую структуру: на первой странице располагается информация о пунктах назначения. В первой столбце располагаются названия пунктов назначения, во втором – координата широты пункта назначения, в третьем – координата долготы, в четвертом – значение индекса пункта назначения. Формат записи координат следующий: grad min, где grad – это значение градуса координаты точки, min – это значение минуты координаты точки. На второй странице файла располагается информация о путевых точках и отрезках дорог.

В таблице 3.1 представлен фрагмент базы данных, содержащий информацию о путевых точках

Таблица 3.1 – Информация о путевых точках

Широта	Долгота	Индекс
52 42.778	41 40.597	0
52 52.904	41 52.591	1
52 53.522	42 01.908	2
52 56.229	42 04.265	3
...	...	...
53 17.969	42 14.109	79
53 17.961	42 16.594	80
53 18.940	42 19.155	81

В таблице 3.2 представлен фрагмент базы данных, содержащий информацию о пунктах назначения.

Таблица 3.2 – Информация о населенных пунктах

Населенный пункт	Широта	Долгота	Индекс
Байловка	53 17.077	42 04.055	1000
Волхонщина	53 14.794	42 5.78	2000
Пичаево	53 14.1	42 12.1	3000
Рудовка	53 6	42 23	4000
Свиньино	53 13.2	42 10.7	5000
М.Гагарино	53 03.3	42 02.5	6000
Б.Шереметьево	53 5.98	42 16.8	7000
Бондарский	52 57.11	42 2.3	8000
Васильево	53 12	42 14	9000
Вернадовка	53 18.285	42 22.121	10000
Заречье	53 15.417	42 10.799	11000
Анненка	53 1.34	42 32.753	12000
Питим	53 12	42 21	13000
Вяжли	53 22	42 10	14000
Осино-Гай	53 1.75	42 24.23	15000
Столовое	52 47.519	41 45.912	16000
Керша	52 52.898	41 52.613	17000
Покрово-Васильевское	53 11.07	42 14.57	18000
Октябрьский	53 15.969	41 59.444	19000
Бондари	52 57	42 04	20000
Куровщино	52 53.4	42 25.08	21000
Максимовка	52 58	42 11	22000
Прибытки	52 54.413	42 3.872	23000
Тамбов	52 43	41 26	24000

В таблице 3.3 представлен фрагмент базы данных, содержащий информацию об участках дороги.

Таблица 3.3 – Информация об отрезках дорог

Индекс1	Индекс2
24000	0
14	17000
14	2
17	18
3	20000

2) Подсистема визуализации – подсистема, необходимая для вывода исходных данных на экран и вывода результата расчета. Исходными данными для данной подсистемы являются координаты пунктов назначения, путевых точек и массив отрезков дорог. Результатом работы данной подсистемы является изображение, содержащее карту с исходными данными либо карту с проложенным маршрутом.

На рисунке 3.1 представлена карта, построенная подсистемой визуализации на основе данных, загруженных из базы данных.

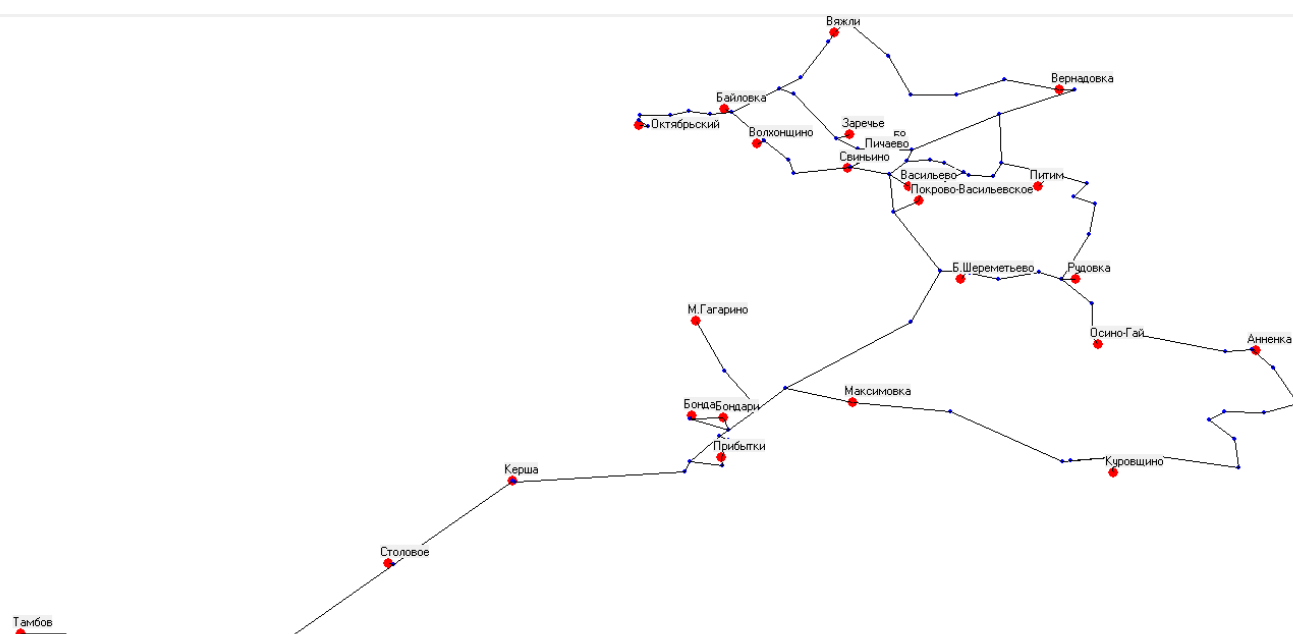


Рисунок 3.1 – Схема расположения населенных пунктов

3) Подсистема расчета кратчайших вариантов проезда между пунктами назначения осуществляет нахождение всех вариантов проезда между всеми пунктами назначения и выбирает кратчайший из них. Входными данными блока являются координаты пунктов назначения, путевых точек и массив отрезков дорог. Выходными данными являются:

- матрица  $E$ , каждый  $E_{ij}$  и  $E_{ji}$  элемент которой содержит наименьшее значение длины пути из  $i$ -го пункта в  $j$ -й;
- таблица дорог, представляющая собой совокупность строк. Первый и последний элемент строки – это индексы пунктов назначения, а остальные элементы строки – это индексы путевых точек, связывающих между собой пункты назначения.

4) Подсистема расчета оптимального порядка перемещения выполняет преобразование матрицы  $E$ . Входными данными данной подсистемы является матрица  $E$ . Выходными данными является массив ребер, образующих замкнутый контур.

5) Блок обработки результатов выполняет преобразование полученных данных для вывода результата на экран. Входными данными являются:

- таблица дорог;
- массив ребер.

Выходными данными является последовательность индексов пунктов назначения и путевых точек. Данная последовательность выводится в виде текста на экран, а также поступает в графическую подсистему для изображения маршрута на карте.

Схематическое представление структурной схемы находится в приложении А.

#### 4 ОПИСАНИЕ СХЕМЫ РАБОТЫ САПР

Чтение из базы данных производится при помощи использования OLE – технологий /4/. Далее производится преобразование абсолютных координат в экранные для отображения карты на экране.

После того как абсолютные координаты были преобразованы, на экран выводится карта при помощи графической подсистемы.

Далее производится процедура поиска всех возможных проездов между  $i$ -м и  $j$ -м пунктами. Каждый из вариантов проезда представляет собой массив индексов путевых точек и пунктов назначения. Первый и последний элементы массива. Все остальные элементы массива содержат индексы путевых точек, последовательное посещение которых приведет нас к одному пункту назначения в другой. Для каждого варианта проезда вычисляется его длина. Для этого необходимо найти сумму расстояний между соседними точками из массива индексов.

Далее из всех вариантов проезда ищется кратчайший. Значение кратчайшего расстояния между  $i$ -м и  $j$ -м пунктами присваивается  $E_{i,j}$  и  $E_{j,i}$  элементам матрицы  $C$ , предварительно заполненной значениями -1. Значение -1 означает, что расстояние между  $i$ -м и  $j$ -м пунктами равно бесконечности.

Затем процедура нахождения кратчайшего варианта проезда между двумя пунктами назначения применяется для каждого пункта, в результате чего происходит заполнение матрицы  $E$ .

После того, как матрица  $E$  была сформирована, выполняется алгоритм вычисления кратчайшего порядка посещения пунктов назначения. В результате работы данного алгоритма был получен массив ребер. Ребро – это пара индексов пунктов назначения. То есть ребро  $(i,j)$  определяет, что из пункта  $a_i$  необходимо перемещаться в пункт  $a_j$ . Все ребра образуют замкнутую последовательность индексов.

Поскольку кратчайшие варианты проезда между пунктами и порядок их посещения были определены, на основе полученных данных формируется

маршрут. Маршрут представляет собой последовательность индексов пунктов назначения и путевых точек. Затем он выводится на экран, а также он отображается на карте при помощи графической подсистемы.

Схематическое представление схемы работы САПР находится в приложении Б.

## 5 ОПИСАНИЕ ВИДОВ ОБЕСПЕЧЕНИЙ

### 5.1 Математическое обеспечение

#### 5.1.1 Перевод координат географического положения в экранные.

Необходимо выполнить преобразование координат долготы и широты точки в координаты, соответствующие диапазону области построения.

Для преобразования координаты долготы необходимо:

- 1) перевести координаты долготы в минуты;
- 2) найти коэффициент  $k$ , рассчитываемый по формуле:

$$k = \frac{(x - x_1)}{(x_2 - x_1)}$$

где  $x_1$  – значение левой границы в минутах;

$x_2$  – значение правой границы в минутах;

$x$  – значение долготы преобразуемой точки в минутах.

- 3) найти экранную координату  $x$  по формуле:

$$x_э = k * n$$

где  $x_э$  – значение экранной координаты для точки;

$n$  – длина области построения в пикселях.

Для преобразования координаты широты необходимо:

- 1) перевести координаты широты в минуты;
- 2) найти коэффициент  $k$ , рассчитываемый по формуле:

$$k = \frac{(y - y_1)}{(y_2 - y_1)}$$



где  $y_1$  – значение нижней границы в минутах;  
 $y_2$  – значение верхней границы в минутах;  
 $y$  – значение широты преобразуемой точки в минутах.

3) найти экранную координату  $y$  по формуле:

$$y_3 = m - k * m$$

где  $y_3$  – значение экранной координаты для точки;  
 $m$  – ширина области построения в пикселях.

### 5.1.2 Нахождение длины отрезка дороги

Отрезок дороги – это отрезок прямой на карте, соединяющий 2 путевые точки, 2 пункта назначения, либо пункт назначения и путевую точку. Таким образом, задача сводится к нахождению расстояния между 2-мя точками. Однако положение точки определяется координатами широты и долготы, поэтому для нахождения длины отрезка дороги в километрах необходимо выполнить следующую последовательность действий:

- 1) перевести координаты широты и долготы обеих точек в минуты;
- 2) перевести координату широты каждой точки в километры по формуле:

$$y = y_{ш} \cdot 1.852, \text{ км}$$

где  $y_{ш}$  – значение координаты широты точки в минутах;  
 $y$  – значение координаты широты точки в километрах.  
 3) перевести координату долготы каждой точки в километры по формуле:

$$x = x_d \cdot \cos(y_{ш}) \cdot 1.852, \text{ км}$$

где  $y_{ш}$  – значение координаты широты точки в минутах;  
 $x$  – значение координаты долготы точки в километрах;  
 $x_d$  – значение координаты долготы точки в минутах.

4) найти расстояние между точками по формуле:

$$s = \sqrt{(x_2 - x_1)^2 +$$

где  $x_1$  – значения координаты долготы первой точки в километрах;  
 $x_2$  – значения координаты долготы второй точки в километрах;

$y_1$  – значения координаты широты первой точки в километрах;

$y_2$  – значения координаты широты второй точки в километрах;

### 5.1.3 Нахождение кратчайшего варианта проезда между пунктами

Задача заключается в нахождении кратчайшего варианта проезда между пунктами. Для решения данной задачи применяется метод полного перебора. Для демонстрации целесообразности применения данного метода необходимо принять некоторые допущения.

#### 5.1.3.1 Принятие допущений

Пусть  $a_i$  и  $a_j$  – пункты назначения,  $V$  – множество путевых точек,  $S_{ij}^r$  – один из множества  $S_{ij}$  вариантов проезда между пунктами  $a_i$  и  $a_j$ ,  $i, j = 1..m$ , (где  $m$  – количество пунктов назначения),  $r = 1..N$  (где  $N$  – количество вариантов проезда между  $a_i$  и  $a_j$ )

Количество всевозможных вариантов соединения путевых точек в случае, если каждая путевая точка связана с  $m-1$  путевой точкой и двумя пунктами назначения равно  $(m+1)!$ . Таким образом, при значении числа путевых точек больше 13 решение задачи данным методом при помощи современной ЭВМ займет около 3 часов. Однако, так как данная задача решается применительно к существующей дорожной ситуации, то совершенно очевидно, что количество вариантов соединения путевых точек отрезками дорог очень невелико. И очень частыми являются случаи, когда путевая точки имеется только 2 связи. Математически данное утверждение это можно представить неравенством:

$$N \ll (m + 1)!$$

В результате принятия данного допущения можно применить метод полного перебора для отыскания кратчайшего варианта проезда между 2-мя пунктами.

#### 5.1.3.2 Описание метода решения задачи

Рассмотрим алгоритм нахождения всех вариантов проезда между двумя пунктами назначения на примере, представленном на рисунке 5.1.

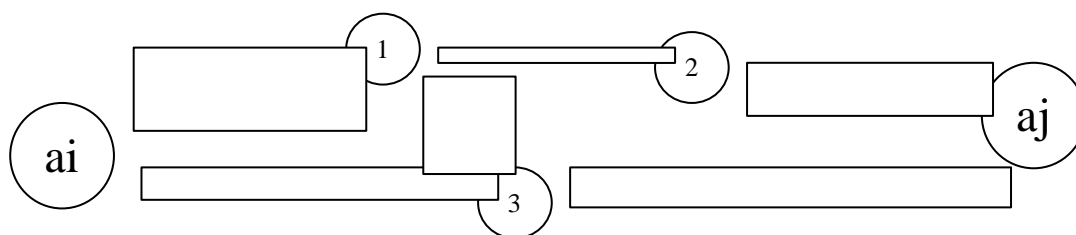


Рисунок 5.1 – Схема расположения точек на карте

На рисунке 5.1 представлены 2 пункта назначения:  $a_i, a_j$ , 3 путевых точки: 1, 2, 3, 6 отрезков дорог:  $\{a_i; 1\}, \{1; 3\}, \{3; a_j\}, \{1; 2\}, \{2; a_j\}, \{a_i; 3\}$ .

На первоначальном этапе строится дерево, корнем которого является пункт отправления (в данном случае пункт  $a_i$ ), «листьями» - путевые точки, а ветвями – отрезки дорог. Существует система правил, которые необходимо соблюдать.

1) Запрещено двигаться из путевой точки в пункт отправления, в путевую точку, уже входящую в данный маршрут или в саму себя. В данном случае вместо значения точки в дереве ставится -1.

2) В случае, если в результате какого-либо обхода был достигнут пункт назначения, вместо значения точки в дереве ставится 0.

3) Процесс етвления продолжается рекурсивно до тех пор, пока каждый элемент множества «листьев» не будет входить в множество  $(-1, 0)$ .

На рисунке 5.2 представлено дерево для исходных данных, представленных на рисунке 5.1.

После завершения рекурсивного процесса в рассмотрение принимаются все ветви, последний элемент которых равен 0. Они образуют множество  $S_{ij}$  всех возможных вариантов проезда между пунктами отправки и назначения. Множество  $S_{ij}$  для представленного на рисунке 3 дерева образуется подмножествами:  $S^1_{ij} = (a_i, 1, 3, a_j)$ ;  $S^2_{ij} = (a_i, 1, 2, a_j)$ ;  $S^3_{ij} = (a_i, 3, a_j)$ ;  $S^4_{ij} = (a_i, 3, 1, 2, a_j)$ .

На следующем этапе находятся длины  $L(S^r_{ij})$  данных вариантов проезда и среди них выбирается такой  $S^r_{ij}$  длина которого минимальна. Длину маршрута можно найти по формуле как сумму длин всех отрезком, входящих в маршрут. В свою очередь длинну отрезка можно рассчитать, если заданы координаты его начала и конца, методом, описанным в разделе 5.1.2.

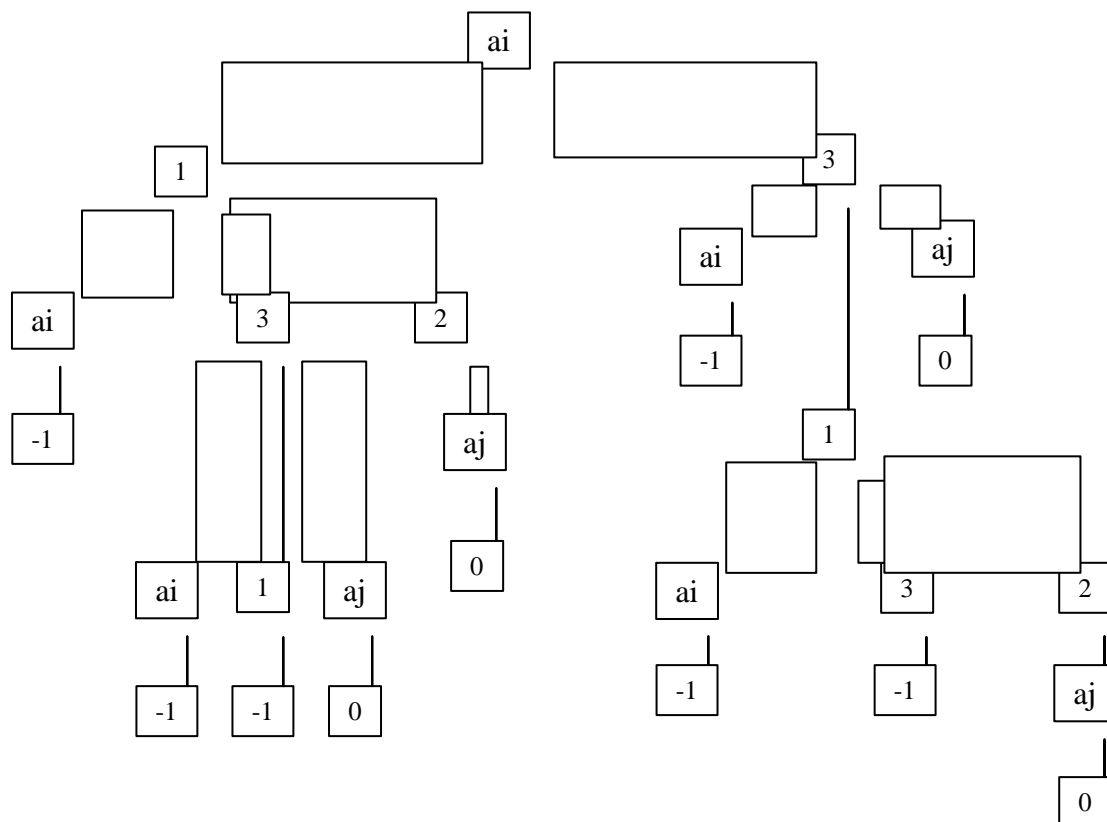


Рисунок 5.2 – Схема дерева

Таким образом, если применить данный алгоритм к каждой паре пунктов назначения, то будут найдены все варианты проезда между всеми пунктами. Кратчайшие варианты проезда образуют матрицу  $E$ . Значение элемента матрицы может быть найдено по формуле:

$$E_{ij} = E_{ji} = L(S_{ij}^{min})$$

где  $E_{ij}$  – элемент матрицы  $E$ ;

$L(S_{ij}^{min})$  – кратчайший вариант проезда между  $i$ -м и  $j$ -м пунктами.

### 5.1.3.3 Анализа метода решения

Основными критериями работы алгоритма в данном случае являются: время работы и точность нахождения результата.

В результате того, что  $N \ll (m+1)!$ , время рассмотрения  $N$  комбинаций гораздо меньше времени рассмотрения  $(m+1)!$  комбинаций. Вследствие этого время выполнения алгоритма становится незначительным.

Поскольку в результате выполнения алгоритма были рассмотрены все варианты проезда, то можно говорить о том, что найденный кратчайший вариант проезда между пунктами назначения действительно является наикратчайшим.

Таким образом, алгоритм удовлетворяет всем требованиям и рассмотрение других методов решения данной задачи бессмысленно, поскольку только метод полного перебора предоставляется гарантированно точное решение. К преимуществам других методов решения данной задачи можно отнести меньшее время, затрачиваемое на работу, однако результаты работы данных алгоритмов являются приблизительными.

#### 5.1.4 Нахождение порядка посещения пунктов назначения

Нахождение порядка однократного посещения всех пунктов с последующим возвратом в начальный пункт также называется задачей коммивояжера. Важным при этом является критерий, на основе которого осуществляется этот поиск. В данной работе учитывается только один критерий – общая длина пути. В данном случае задача коммивояжера называется симметричной. Иными словами, если  $L(i,j)$  – это есть расстояние между пунктами с индексами  $i$  и  $j$ , то справедливо следующее:

$$L(i, j) = L(j, i)$$

Постановка задачи коммивояжера была описана в разделе 2.3. С точки зрения теории графов задача коммивояжера формулируется так: дан полный граф с  $n$  вершинами, длина ребра  $(i,j) = E_{ij}$ . Найти в данном графе гамильтонов цикл минимальной длины /3/.

Полный граф – простой граф, в котором каждая пара различных вершин смежна. На рисунке 5.3 представлен пример полного графа.

Гамильтоновым циклом называется гамильтонов путь, начальная и конечная вершины которого совпадают. Гамильтонов путь (или гамильтонова цепь) – путь (цепь), содержащий каждую вершину графа ровно один раз.

Существует множество методов решения задачи коммивояжера. К ним относятся: метод полного перебора, «жадный» алгоритм, «деревянный» алгоритм, метод «ветвей и границ», алгоритм Дейкстры.

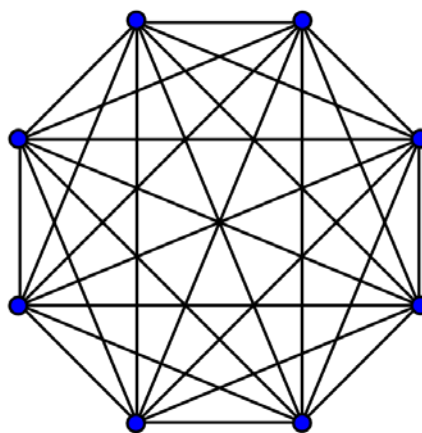


Рисунок 5.3 – Полный граф

#### 5.1.4.1 Метод полного перебора

Также данный метод имеет названия: перебор в лексикографическом порядке, перебор животной силой. Суть метода заключается в том, что необходимо рассмотреть все возможные гамильтоновы циклы в графе. Иными словами, если дан полный симметричный граф, количество вершин которого равно  $n$ . Тогда количество всех гамильтоновых циклов, содержащихся в данном графе равно:

$$m = \frac{n!}{2}$$

где  $n$  – количество вершин в графе;

$m$  – количество гамильтоновых циклов в графе.

Таким образом необходимо найти длины всех  $m$  гамильтоновых циклов и среди них выбрать цикл с наименьшей длиной. Каждый гамильтонов цикл рассматривается единственный раз и все циклы отличны друг от друга.

Главным недостатком данного метода является большой объем вычислений, производимых при поиске решения. Например, если дан граф с количеством вершин  $n$  равным 15, то количество всех гамильтонов циклов данного графа сопоставимо с числом  $10^{12}$ . То есть необходимо вычислить длину каждого из 15! возможных гамильтоновых циклов и среди них выбрать кратчайший. Таким образом, время выполнения алгоритма в данном случае можно вычислить по формуле:

$$T = t \cdot \frac{15!}{2}$$

где  $T$  – время выполнения алгоритма;

$t$  – время расчета длины одного гамильтонова цикла.

Решение данной задачи методов полного перебора займет примерно 3 дня, что абсолютно недопустимо, поскольку время работы всей проектируемой системы должно быть не более 5 минут. Таким образом, данный алгоритм применим только к задачам, количество пунктов назначения в которых  $<10$ . То есть, вероятность применения данного метода в реальных условиях крайне мала, так как в среднем количество пунктов назначения, которые необходимо посетить, более 15.

#### 5.1.4.2 «Жадный» алгоритм

Жадный алгоритм – алгоритм нахождения наикратчайшего расстояния путём выбора самого короткого, ещё не выбранного ребра, при условии, что оно не образует цикла с уже выбранными рёбрами. «Жадным» этот алгоритм назван потому, что на последних шагах приходится жестоко расплачиваться за жадность.

Посмотрим, как поведет себя при решении задачи коммивояжера жадный алгоритм. Здесь он превратится в стратегию «иди в ближайший (в который еще не входил) город». Жадный алгоритм, очевидно, бессилен в этой задаче. На рисунке 5.4 изображен пример графа.

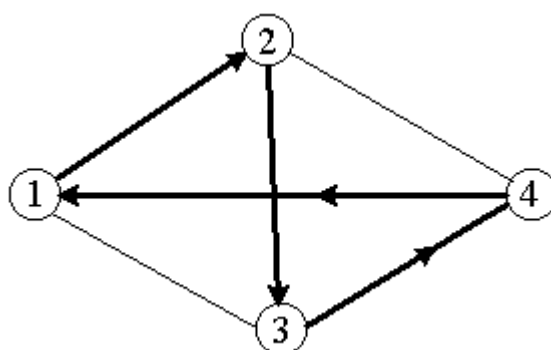


Рисунок 5.4 – Результат работы «жадного» алгоритма

Пусть коммивояжер стартует из города 1. Алгоритм «иди в ближайший город» выведет его в город 2, затем 3, затем 4. На последнем шаге придется

платить за жадность, возвращаясь по длинной диагонали ромба. В результате получится не кратчайший, а длиннейший тур.

В пользу процедуры «иди в ближайший» можно сказать лишь то, что при старте из одного города она не уступит стратегии «иди в дальнейший».

Как видим, жадный алгоритм ошибается. Можно ли доказать, что он ошибается умеренно, что полученный им тур хуже минимального, положим, в 1000 раз? Мы докажем, что этого доказать нельзя, причем не только для жадного логарифма, а для алгоритмов гораздо более мощных. Но сначала нужно договориться, как оценивать погрешность неточных алгоритмов, для определенности, в задаче минимизации.

Пусть  $f_B$  – настоящий минимум, а  $f_A$  – тот квазiminимум, который получен по алгоритму. Тогда справедливо равенство:

$$\frac{f_A}{f_B} = 1 + \varepsilon$$

где  $\varepsilon$  – мера погрешности.

Если степень вершины нечетна – то в ней линия должна начинаться или кончиться. На рисунке 5.5 вершин нечетной степени две: в одной линия начинается, в другой – кончается. Однако на рисунке 5.6 имеется четыре вершины степени три, но у одной линии не может быть четыре конца. Если же нужно прочертить фигуру одной замкнутой линией, то все ее вершины должны иметь четную степень.

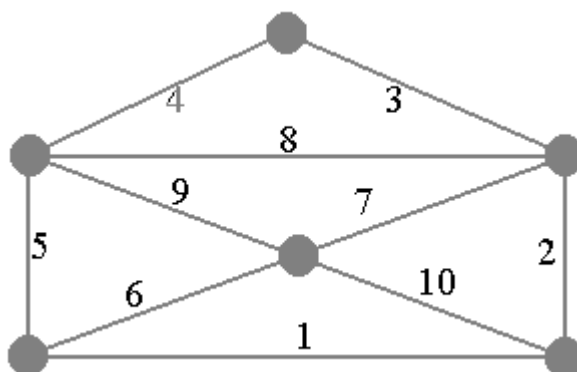


Рисунок 5.5 – Граф с двумя вершинами нечетной степени



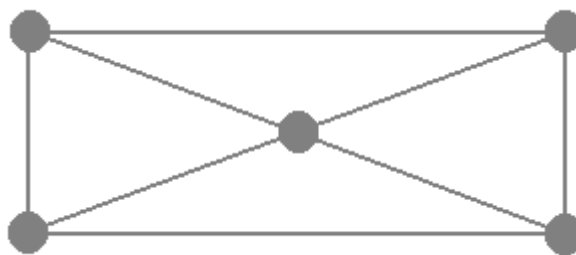


Рисунок 5.6 – Граф с четырьмя вершинами нечетной степени

Верно и обратное утверждение: если все вершины имеют четную степень, то фигуру можно нарисовать одной незамкнутой линией. Действительно, процесс проведения линии может кончиться, только если линия придет в вершину, откуда уже выхода нет: все ребра, присоединенные к этой вершине (обычно говорят: инцидентные этой вершине), уже прочерчены. Если при этом нарисована вся фигура, то нужное утверждение доказано; если нет, удалим уже нарисованную часть  $G'$ . После этого от графа останется одна или несколько связных компонент; пусть  $G'$  – одна из таких компонент. В силу связности исходного графа  $G$ ,  $G'$  и  $G''$  имеют хоть одну общую вершину, скажем,  $v$ . Если в  $G''$  удалены какие-то ребра, то по четному числу от каждой вершины. Поэтому  $G''$  – связный и все его вершины имеют четную степень. Построим цикл в  $G''$  (может быть, не нарисовав всего  $G''$ ) и через  $v$  добавим прорисованную часть  $G''$  к  $G'$ . Увеличивая таким образом прорисованную часть  $G'$ , мы добьемся того, что  $G'$  охватит весь  $G$ .

Эту задачу когда-то решил Эйлер, и замкнутую линию, которая покрывает все ребра графа, теперь называю эйлеровым циклом. По существу была доказана следующая теорема: эйлеров цикл в графе существует тогда и только тогда, когда:

- 1) граф связный;
- 2) все его вершины имеют четные степени.

#### 5.1.4.3 «Деревянный» алгоритм

Теперь можно обсудить алгоритм решения задачи коммивояжера через построение кратчайшего остовного дерева. Для краткости будет называть этот алгоритм деревянным.

Вначале обсудим свойство спрямления. Рассмотрим какую-нибудь цепь, например, представленную на рисунке 5.7.

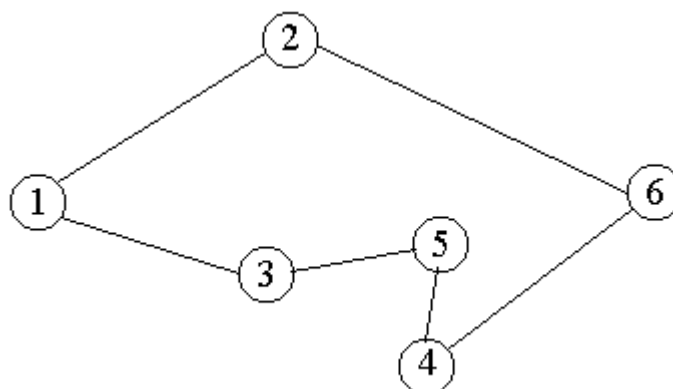


Рисунок 5.7 – Цепь

Если справедливо неравенство треугольника, то:

$$d[1,3] \leq d[1,2] + d[2,3]$$

$$d[3,5] \leq d[3,4] + d[4,5]$$

Сложив эти два неравенства, получим:

$$d[1,3] + d[3,5] \leq d[1,2] + d[2,3] + d[3,4] + d[4,5]$$

По неравенству треугольника получим:

$$d[1,5] \leq d[1,3] + d[3,5]$$

Окончательно:

$$d[1,5] \leq d[1,2] + d[2,3] + d[3,4] + d[4,5]$$

Итак, если справедливо неравенство треугольника, то для каждой цепи верно, что расстояние от начала до конца цепи меньше (или равно) суммарной длины всех ребер цепи. Это обобщение расхожего убеждения, что прямая короче кривой.

«Деревянный» алгоритм представляет собой последовательное выполнение следующих пунктов.

1) Построим на входном графе задачи коммивояжера кратчайшее остовное дерево и удвоим все его ребра. Получим граф  $G$  – связный и с вершинами, имеющими только четные степени.

2) Построим эйлеров цикл  $G$ , начиная с вершины 1, цикл задается перечнем вершин.

3) Просмотрим перечень вершин, начиная с 1, и будем зачеркивать каждую вершину, которая повторяет уже встреченную в последовательности. Останется тур, который и является результатом алгоритма.

Теорема. Погрешность деревянного алгоритма равна 1.

Доказательство. Возьмем минимальный тур длины  $f_B$  и удалим из него максимальное ребро. Длина получившейся гамильтоновой цепи  $L_{HC}$  меньше  $f_B$ . Но эту же цепь можно рассматривать как остовное дерево, т. к. эта цепь достигает все вершины и не имеет циклов. Длина кратчайшего остовного дерева  $L_{MT}$  меньше или равна  $L_{HC}$ . Имеем цепочку неравенств:

$$f_B > L_{HC} \geq L_{MT}$$

Но удвоенное дерево – оно же эйлеров граф – мы свели к туру посредством спрямлений, следовательно, длина полученного по алгоритму тура удовлетворяет неравенству

$$2L_{MT} > f_A$$

Получаем цепочку неравенств:

$$2f_B > 2L_{HC} \geq 2L_{MT} \geq f_A$$

Таким образом, из неравенства:

$$\frac{f_A}{f_B} > 1 + \varepsilon$$

получаем, что:

$$\varepsilon = 1$$

Теорема доказана.

Таким образом, мы доказали, что «деревянный» алгоритм ошибается менее, чем в два раза. Такие алгоритмы уже называют приближительными, а не просто эвристическими.

#### 5.1.4.4 Метод «ветвей и границ»

К идее метода ветвей и границ приходили многие исследователи, но Литтл с соавторами на основе указанного метода разработали удачный алгоритм решения задачи коммивояжера и тем самым способствовали популяризации подхода. С тех пор метод ветвей и границ был успешно применен ко многим задачам, для решения задачи коммивояжера было придумано несколько других модификаций метода, но в большинстве учебников излагается пионерская работа Литтла.

Общая идея тривиальна: нужно разделить огромное число перебираемых вариантов на классы и получить оценки (снизу – в задаче минимизации, сверху – в задаче максимизации) для этих классов, чтобы иметь возможность отбрасывать варианты не по одному, а целыми классами. Трудность состоит в том, чтобы найти такое разделение на классы (ветви) и такие оценки (границы), чтобы процедура была эффективной.

Изложим алгоритм Литтла на примере матрицы  $E$ , представленной в таблице 5.1.

Таблица 5.1 – Исходная матрица  $E$

	1	2	3	4	5	6
1	-1	6	4	8	7	14
2	6	-1	7	11	7	10
3	4	7	-1	4	3	10
4	8	11	4	-1	5	11
5	7	7	3	5	-1	7
6	14	10	10	11	7	-1

Нам будет удобнее трактовать  $E_{ij}$  как стоимость проезда из города  $i$  в город  $j$ . Допустим, что добрый мэр города  $j$  издал указ выплачивать каждому въехавшему в город коммивояжеру 5 долларов. Это означает, что любой тур подешевеет на 5 долларов, поскольку в любом туре нужно въехать в город  $j$ . Но поскольку все туры равномерно подешевели, то прежний минимальный тур будет и теперь стоить

меньше всех. Добрый же поступок мэра можно представить как уменьшение всех чисел  $j$ -го столбца матрицы  $E$  на 5. Если бы мэр хотел спроводить коммивояжеров из  $j$ -го города и установил награду за выезд в размере 10 долларов, это можно было бы выразить вычитанием 10 из всех элементов  $j$ -й той строки. Это снова бы изменило стоимость каждого тура, но минимальный тур остался бы минимальным. Итак, доказана следующая лемма.

Вычитая любую константу из всех элементов любой строки или столбца матрицы  $C$ , мы оставляем минимальный тур минимальным.

Для алгоритма нам будет удобно получить побольше нулей в матрице  $E$ , не получая там, однако, отрицательных чисел. Выполним процедуру приведения матрицы  $E$  по строкам. Для этого из каждой строки вычитается ее минимальный элемент. В таблице 5.2 представлена матрица  $E$ , приведенная по строкам.

Таблица 5.2 – Матрица  $E$ , приведенная по строкам

	1	2	3	4	5	6	min
1	-1	2	0	4	3	10	4
2	0	-1	1	5	1	4	6
3	1	4	-1	1	0	7	3
4	4	7	0	-1	1	7	4
5	4	4	0	2	-1	4	3
6	7	3	3	4	0	-1	7

Затем производится приведение матрицы по столбцам. Для этого из каждого столбца матрицы, приведенной по строкам, вычитается его минимальный элемент. В таблице 5.3 представлена матрица  $E$ , приведенная по строкам и столбцам.

Значения -1 по диагонали означают, что перемещение из пункта  $i$  в пункт  $i$  невозможно. Заметим, что сумма констант приведения по строкам равна 27, сумма по столбцам 7, сумма сумм равна 34.

Таблица 5.3 – Матрица E, приведенная по строкам и столбцам

	1	2	3	4	5	6
1	-1	0	0	3	3	6
2	0	-1	1	4	1	0
3	1	2	-1	0	0	3
4	4	5	0	-1	1	3
5	4	2	0	1	-1	1
6	7	1	3	3	0	-1
min	0	2	0	1	0	4

Тур можно задать системой из шести подчеркнутых (выделенных другим цветом) элементов матрицы E, например, такой, как показано на табл. 2. Подчеркивание элемента означает, что в туре из  $i$ -го элемента идут именно в  $j$ -тый. Для тура из шести городов подчеркнутых элементов должно быть шесть, так как в туре из шести городов есть шесть ребер. Каждый столбец должен содержать ровно один подчеркнутый элемент (в каждый город коммивояжер въехал один раз), в каждой строке должен быть ровно один подчеркнутый элемент (из каждого города коммивояжер выехал один раз); кроме того, подчеркнутые элементы должны описывать один тур, а не несколько меньших циклов. Сумма чисел подчеркнутых элементов есть стоимость тура.

Если в матрице, представленной таблицей 5.3, удастся построить правильную систему подчеркнутых элементов, т.е. систему, удовлетворяющую трем вышеописанным требованиям, и этими подчеркнутыми элементами будут только нули, то ясно, что для этой матрицы мы получим минимальный тур. Но он же будет минимальным и для исходной матрицы E, только для того, чтобы получить правильную стоимость тура, нужно будет обратно прибавить все константы приведения, и стоимость тура изменится с 0 до 34. Таким образом, минимальный тур не может быть меньше 34. Мы получили оценку снизу для всех туров.

Теперь приступим к ветвлению. Для этого сделаем шаг оценки нулей. Рассмотрим нуль в клетке (1,2) приведенной матрицы. Он означает, что цена перехода из города 1 в город 2 равна 0. А если мы не пойдём из города 1 в город 2? Тогда все равно нужно въехать в город 2 за цены, указанные во втором столбце; дешевле всего за 1 (из города 6). Далее, все равно надо будет выехать из города 1 за цену, указанную в первой строке; дешевле всего в город 3 за 0. Суммируя эти два минимума, имеем  $1+0=1$ : если не ехать «по нулю» из города 1 в город 2, то надо заплатить не меньше 1. Это и есть оценка нуля. Оценки всех нулей поставлены в таблице 5.4 правее и выше нуля (оценки нуля, равные нулю, не ставились).

Таблица 5.4 – Степени нулей

	1	2	3	4	5	6
1	-1	0 <sup>1</sup>	0	3	3	6
2	0 <sup>1</sup>	-1	1	4	1	0
3	1	2	-1	0 <sup>1</sup>	0	3
4	4	5	0 <sup>1</sup>	-1	1	3
5	4	2	0	1	-1	0
6	7	1	3	3	0 <sup>1</sup>	-1

Выберем максимальную из этих оценок (в примере есть несколько оценок, равных единице, выберем первую из них, в клетке (1,2)).

Итак, выбрано нулевое ребро (1,2). Разобьём все туры на два класса – включающие ребро (1,2) и не включающие ребро (1,2). Про второй класс можно сказать, что придется приплатить еще 1, так что туры этого класса стоят 35 или больше.

Что касается первого класса, то в нем надо рассмотреть матрицу из таблицы 5.5 с вычеркнутой первой строкой и вторым столбцом.

Таблица 5.5 – Преобразованная матрица Е

	1	3	4	5	6
2	0 <sup>1</sup>	1	4	1	0
3	1	-	0 <sup>1</sup>	0	3
4	4	0 <sup>1</sup>	-	1	3
5	4	0	1	-	0
6	7	3	3	0 <sup>1</sup>	-

Дополнительно в уменьшенной матрице поставлен запрет в клетке (2,1), так как выбрано ребро (1,2) и замыкать преждевременно тур ребром (2,1) нельзя. Уменьшенную матрицу можно привести на 1 по первому столбцу, так что каждый тур, ей отвечающий, стоит не меньше 35. Результат ветвлений и получения оценок показан на рисунке 5.8.

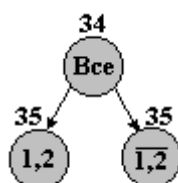


Рисунок 5.8 – Начальная стадия ветвления

Кружки представляют классы: верхний кружок – класс всех туров; нижний левый – класс всех туров, включающих ребро (1,2); нижний правый – класс всех туров, не включающих ребро (1,2). Числа над кружками – оценки снизу.

Продолжим ветвление в положительную сторону: влево - вниз. Для этого оценим нули в уменьшенной матрице приведенной матрице, представленной на таблице 5.6.

Максимальная оценка в клетке (3,1) равна 3. Таким образом, оценка для правой нижней вершины есть 38. Для оценки левой нижней вершины нужно вычеркнуть из матрицы преобразованной матрицы Е еще строку 3 и столбец 1, получив новую матрицу Е, представленную в таблице 5.7.



Таблица 5.6 – Степени нулей преобразованной приведенной матрицы E

	1	3	4	5	6
2	$0^1$	1	4	1	0
3	$0^3$	-1	$0^1$	0	3
4	3	$0^1$	-1	1	3
5	3	0	1	-1	0
6	6	3	3	$0^1$	-1

Таблица 5.7 – Второй этап преобразования матрицы E

	3	4	5	6
2	1	4	1	0
4	$0^1$	-1	1	3
5	0	1	-1	0
6	3	3	$0^1$	-1

В эту матрицу нужно поставить запрет в клетку (2,3), так как уже построен фрагмент тура из ребер (1,2) и (3,1), т.е. [3,1,2], и нужно запретить преждевременное замыкание (2,3). Эта матрица приводится по столбцу на 1 (таким образом, каждый тур соответствующего класса (т.е. тур, содержащий ребра (1,2) и (3,1)) стоит 36 и более. Преобразованная на данном этапе матрица представлена в таблице 5.8.

Таблица 5.8 – Приведенная матрица E, преобразованная на втором этапе

	3	4	5	6
2	1	3	1	0
4	$0^1$	-1	1	3
5	0	$0^2$	-1	0
6	3	2	$0^3$	-1

Дерево, полученное на втором этапе ветвления изображено на рисунке 5.9

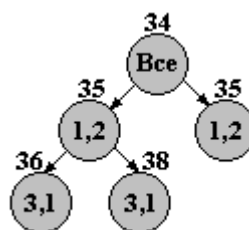


Рисунок 5.9 – Второй этап ветвления

Оцениваем теперь нули в приведенной матрице Е нуль с максимальной оценкой 3 находится в клетке (6,5). Отрицательный вариант имеет оценку 41. Для получения оценки положительного варианта убираем строчку 6 и столбец 5, ставим запрет в клетку (5,6). Преобразованная на данном этапе матрица представлена в таблице 5.9. Эта матрица неприводима.

Таблица 5.9 – Третий этап преобразования матрицы Е

	3	4	6
2	1	3	$0^3$
4	$0^3$	-1	3
5	0	$0^3$	0

Следовательно, оценка положительного варианта не увеличивается. Дерево ветвлений, полученное в ходе преобразования матрицы и оценки нижней границы, представлено на рисунке 5.10.

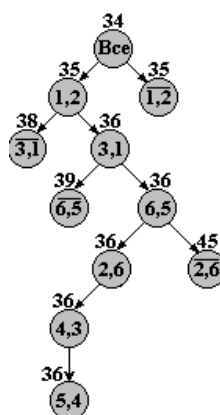


Рисунок 5.10 – Конечный результат ветвления

Оценивая нули в матрице на таблице 5.9, получаем ветвление по выбору ребра (2,6), отрицательный вариант получает оценку 39, а для получения оценки положительного варианта вычеркиваем вторую строку и шестой столбец, получая новую матрицу. Матрица, полученная на данном этапе представлена таблицей 5.10.

Таблица 5.10 – Конечный этап преобразования матрицы E

	3	4
4	0	-1
5	0	0

В матрицу надо добавить запрет в клетку (5,3), ибо уже построен фрагмент тура [3,1,2,6,5] и надо запретить преждевременный возврат (5,3). Теперь, когда осталась матрица 2x2 с запретами по диагонали, достраиваем тур ребрами (4,3) и (5,4). Мы не зря ветвились, по положительным вариантам. Сейчас получен тур:  $1 \rightarrow 2 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 1$  стоимостью в 36. При достижении низа по дереву перебора класс туров сузился до одного тура, а оценка снизу превратилась в точную стоимость.

Удовлетворительных теоретических оценок быстродействия алгоритма Литтла и родственных алгоритмов нет, но практика показывает, что на современных ЭВМ они часто позволяют решить задачи коммивояжера с  $n = 100$ . Это огромный прогресс по сравнению с полным перебором. Кроме того, алгоритмы типа ветвей и границ являются, если нет возможности доводить их до конца, эффективными эвристическими процедурами /2/.

#### 5.1.4.5 Алгоритм Дейкстры

Одним из вариантов решения задачи коммивояжера является вариант нахождения кратчайшей цепи, содержащей все города. Затем полученная цепь дополняется начальным городом – получается искомый тур.

Можно предложить много процедур решения этой задачи, например, физическое моделирование. На плоской доске рисуется карта местности, в города,

лежащие на развилке дорог, вбиваются гвозди, на каждый гвоздь надевается кольцо, дороги укладываются верёвками, которые привязываются к соответствующим кольцам. Чтобы найти кратчайшее расстояние между  $i$  и  $k$ , нужно взять  $i$  в одну руку и  $k$  в другую и растянуть. Те верёвки, которые натянутся и не дадут разводить руки шире и образуют кратчайший путь между  $i$  и  $k$ . Однако математическая процедура, которая промоделирует эту физическую, выглядит очень сложно. Известны алгоритмы попроще. Один из них – алгоритм Дейкстры, предложенный Дейкстрой ещё в 1959г.

Этот алгоритм решает общую задачу: в ориентированном, неориентированном или смешанном (т. е. такой, где часть дорог имеет одностороннее движение) графе найти кратчайший путь между двумя заданными вершинами.

Пусть число вершин в графе равно  $n$ . Алгоритм использует три массива из  $n$  чисел каждый. Первый массив  $a$  содержит метки с двумя значениями: 0 (вершина ещё не рассмотрена) и 1 (вершина уже рассмотрена); второй массив  $b$  содержит расстояния – текущие кратчайшие расстояния от  $v_i$  до соответствующей вершины; третий массив  $c$  содержит номера вершин –  $k$ -й элемент  $c_k$  есть номер предпоследней вершины на текущем кратчайшем пути из  $v_i$  в  $v_k$ . Матрица расстояний  $E_{ik}$  задаёт длины дуг  $e_{ik}$ ; если такой дуги нет, то  $e_{ik}$  присваивается большое число  $B$ , равное «машинной бесконечности».

Теперь можно описать алгоритм Дейкстры. Алгоритм представляет собой последовательное выполнение следующих действий.

1) Инициализация. В цикле от 1 до  $n$  заполнить нулями массив  $a$ . Заполнить числом  $i$  массив  $c$ . Перенести  $i$ -тую строку матрицы  $E$  в массив  $b$ .

$$a_i = 1$$

$$c_i = 0$$

где  $i$  – номер стартовой вершины.

2) Общий шаг. Найти минимум среди неотмеченных (т. е. тех  $k$ , для которых  $a_k = 0$ ). Пусть минимум достигается на индексе  $j$ , то есть:

$$b_j \leq b_k$$

$$a_j=1$$

если

$$b_k > b_j + d_{jk}$$

то

$$b_k = b_j + d_{jk}$$

$$c_k = j$$

Данное условие означает, что путь  $v_i..v_k$  длиннее, чем путь  $v_i..v_j, v_k$ . Если все  $a_k$  отмечены, то длина пути  $v_i..v_k$  равна  $b_k$ . Теперь надо перечислить вершины, входящие в кратчайший путь.

3) Выдача ответа.

Для выполнения алгоритма нужно  $n$  раз просмотреть массив  $b$  из  $n$  элементов, то есть алгоритм Дейкстры имеет квадратичную сложность.

Проиллюстрируем работу алгоритма Дейкстры численным примером. Для большей сложности, считаем, что некоторые города (вершины)  $i, j$  не соединены между собой, то есть:

$$D[i, j] = \infty$$

На рисунке 5.11 изображена схема расположения пунктов со значениями ребер.

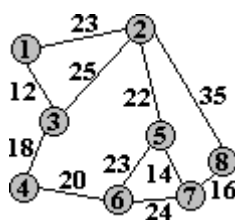


Рисунок 5.11 – Схема расположения пунктов

Изображенному на рисунке 5.8 графу соответствует матрица, представленная таблицей 5.11.

Таблица 5.11 – Исходная матрица

	1	2	3	4	5	6	7	8
1	0	23	12	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
2	23	0	25	$\infty$	22	$\infty$	$\infty$	35
3	12	25	0	18	$\infty$	$\infty$	$\infty$	$\infty$
4	$\infty$	$\infty$	18	0	$\infty$	20	$\infty$	$\infty$
5	$\infty$	22	$\infty$	$\infty$	0	23	14	$\infty$
6	$\infty$	$\infty$	$\infty$	20	23	0	24	$\infty$
7	$\infty$	$\infty$	$\infty$	$\infty$	14	24	0	16
8	$\infty$	35	$\infty$	$\infty$	$\infty$	$\infty$	16	0

Пусть, требуется найти кратчайшие пути из вершины 3. Содержимое массивов a,b,c после выполнения первого пункта представлено в таблице 5.12

Таблица 5.12 – Содержимое массивов на первом шаге

	1	2	3	4	5	6	7	8
a	0	0	1	0	0	0	0	0
b	12	25	0	18	$\infty$	$\infty$	$\infty$	$\infty$
c	3	3	0	3	3	3	3	3

Очевидно, содержимое таблицы меняется по мере выполнения общего шага. Это видно из таблицы 5.13.

Таким образом, для решения задачи коммивояжера нужно  $n$  раз применить алгоритм Дейкстры. Возьмём произвольную пару вершин  $j,k$ . Исключим непосредственное ребро  $C[j,k]$ . С помощью алгоритма Дейкстры найдём кратчайшее расстояние между городами  $j..k$ . Пусть это расстояние включает некоторый город  $m$ . Имеем часть тура  $j,m,k$ . Теперь для каждой пары соседних городов (в данном примере – для  $j,m$  и  $m,k$ ) удалим соответственное ребро и

найдем кратчайшее расстояние. При этом в кратчайшее расстояние не должен входить уже использованный город.

Таблица 5.13 – Изменение содержимого массивов

		1	2	3	4	5	6	7	8
min $b_k=12$	a	1	0	1	0	0	0	0	0
	b	12	25	0	18	$\infty$	$\infty$	$\infty$	$\infty$
	c	3	3	0	3	3	3	3	3
min $b_k=18$	a	1	0	1	1	0	0	0	0
	b	12	25	0	18	$\infty$	38	$\infty$	$\infty$
	c	3	3	0	3	3	4	3	3
min $b_k=25$	a	1	1	1	1	0	0	0	0
	b	12	25	0	18	47	38	$\infty$	60
	c	3	3	0	3	2	4	3	2
min $b_k=38$	a	1	1	1	1	0	1	0	0
	b	12	25	0	18	47	38	62	60
	c	3	3	0	3	2	4	6	2
min $b_k=47$	a	1	1	1	1	1	1	0	0
	b	12	25	0	18	47	38	61	60
	c	3	3	0	3	2	4	5	2
min $b_k=60$	a	1	1	1	1	1	1	0	1
	b	12	25	0	18	47	38	61	60
	c	3	3	0	3	2	4	5	2

Далее аналогично находим кратчайшее расстояние между парами вершин алгоритмом Дейкстры, до тех пор, пока все вершины не будут задействованы. Соединим последнюю вершину с первой и получим тур. Чаще всего это последнее

ребро оказывается очень большим, и тур получается с погрешностью, однако алгоритм Дейкстры можно отнести к приближённым алгоритмам.

#### 5.1.4.6 Анализ методов решения

Для решения задачи в данной работе используется метод ветвей и границ, как наиболее эффективный и быстродейственный метод. При количестве пунктов назначения равном 100, время выполнения алгоритма составляет 1,2 секунды. Метод полного перебора для решения данной задачи не подходит, так как при количестве пунктов назначения равном 13 решение занимает более 3 часов, а такое количество времени для расчета одного маршрутного листа неприемлемо. Деревянный метод был мною исключен из рассмотрения как приблизительный. Очень велика вероятность получения результата примерно в 2 раза хуже действительного. Несмотря на то, что «жадный» алгоритм является одним из наиболее легких в реализации и быстродейственным, очень велика вероятность получения результата, значительно худшего чем действительный.

#### 5.2 Информационное обеспечение

Информационное обеспечение — создание информационных условий функционирования системы, обеспечение необходимой информацией, включение в систему средств поиска, получения, хранения, накопления, передачи, обработки информации, организация банков данных. Создание информационного обеспечения — неременное условие построения и функционирования автоматизированных систем управления.

Наибольшее распространение получила реляционная модель, вследствие простой формы представления данных и развитому теоретическому аппарату, позволяющему легко описывать различные преобразования над ними. Основу реляционной модели составляет совокупность данных, сформированных в виде таблицы. Такая форма представления информации привычна для специалиста, пользующегося различной справочной литературой, поэтому в данной САПР используется именно реляционная модель БД.

В разработанной автоматизированной системе используется только одна база данных «database.xls». Она содержит 3 таблицы:



1) таблица пунктов назначения. В ней содержатся данные о населенных пунктах, в которых располагаются клиенты. Данная таблица содержит поля:

- название населенного пункта;
- координата широты;
- координата долготы.
- индекс пункта назначения

Название населенного пункта представляет собой массив символов. Координаты широты и долготы имеют пользовательский тип данных `coords`. Этот тип данных имеет структуру:

- градусы(`grad`);
- минуты(`min`).

`grad` – переменная целого типа, `min` – вещественная переменная. Индекс пункта назначения – это целое число. Все населенные пункты проиндексированы и имеют индексы 1000,2000 и так далее.

2) таблица путевых точек. В ней содержатся данные о путевых точках. Данная таблица содержит поля:

- координата широты;
- координата долготы;
- индекс путевой точки.

Координаты широты и долготы имеют пользовательский тип `coords`. Индекс путевой точки – это целое число. Все путевые точки проиндексированы и имеют индексы 0,1,...,999.

3) таблица отрезков дорог. В ней содержатся данные об отрезках дорог, соединяющих путевые точки, пункты назначения, либо путевую точку с пунктом назначения. Данная таблица содержит поля:

- индекс 1;
- индекс 2.

Индекс 1 и индекс 2 – это целые числа. Это как раз и есть индексы путевых точек и пунктов назначения. Если в данной таблице содержится запись вида

{индекс1, индекс2}, то это значит, что отрезок дороги соединяет пункт назначения или путевую точку с индексом 1 с путевой точкой или пунктом назначения с индексом 2.

Для начала работы системы необходима загрузка всех таблиц базы данных. Схематическое представление даталогической модели САПР находится в приложении В.

### 5.3 Лингвистическое обеспечение

Целью разработки лингвистического обеспечения является осуществление взаимодействия между пользователем и системой путем применения различных языков описания и управления, интерфейсов.

В настоящее время с развитием графических средств ЭВМ, становится возможным разработка интуитивно-понятных диалоговых средств, чтобы неподготовленный пользователь мог самостоятельно работать с системой, не пользуясь посторонней помощью, а опытный, не испытывал затруднений при использовании всех функций системы и мог настраивать диалоговую систему “под себя”. Такая система называется графическим пользовательским интерфейсом GUI (“Graphical User Interface”) и характеризуется следующими свойствами:

- графический режим работы;
- представление ряда объектов пиктограммами;
- многооконный интерфейс;
- использование манипулятора мыши;
- адекватность изображения на экране изображаемому объекту;
- наглядность;
- стандартизация основных действий и элементов.

Основные средства взаимодействия человека и машины – это различные диалоговые системы. Разработанная автоматизированная система использует тип диалога «заполнение бланков». То есть в поле ввода необходимо ввести имя базы данных, из которой произойдет загрузка данных.

На рисунке 5.12 представлен пример диалога типа «заполнение бланков».



Рисунок 5.12 – «Заполнение бланков» для загрузки базы данных

Первоначально все данные из базы данных заносятся в массивы, используемые системой для выполнения расчетов и преобразования. Обратный контакт с базой данных отсутствует. То есть взаимодействие системы с базой данных заключается только в загрузке данных из базы данных. Все изменения в базе данных производятся в при помощи другой системы.

#### 5.4 Программное обеспечение

Программное обеспечение – совокупность программ, представленных в заданной форме, вместе с необходимой программной документацией, автоматизирующая процесс решения задач проектирования на заданных (выбранных) технических средствах и предназначенная для использования в САПР. От свойств программного обеспечения в значительной мере зависят возможности и показатели эффективности САПР.

Для разрабатываемой САПР необходимо следующее общесистемное программное обеспечение:

- 1) операционная система Microsoft Windows 7;
- 2) среда визуального программирования Delphi 7.0;
- 3) MS Word 2007.

Прикладное программное обеспечение представлено одним программным модулем, в котором выполняется:

- преобразование исходных данных;
- взаимодействие с базой данных;
- математические расчеты;
- решение задачи оптимизации;
- диалог пользователя с ЭВМ.

### 5.5 Техническое обеспечение

Выбор технических средств зависит от требований, предъявляемых к ним. К техническому обеспечению САПР предъявляются следующие требования:

- удобство использования инженерами-проектировщиками, возможность оперативного взаимодействия инженеров с ЭВМ;
- достаточная производительность и объем оперативной памяти ЭВМ для решения задач всех этапов проектирования за приемлемое время;
- возможность одновременной работы с техническими средствами необходимого числа пользователей для эффективной деятельности всего коллектива разработчиков;
- открытость комплекса технических средств для расширения и модернизации системы по мере совершенствования и развития техники;
- высокая надежность, приемлемая стоимость и т.д.

Удовлетворение перечисленных требований возможно только в условиях организации технического обеспечения в виде специализированной ВС, допускающей функционирование в нескольких режимах. Такое техническое обеспечение называют комплексом технических средств САПР (комплексом ТС).

Минимальная конфигурация ЭВМ для работы системы:

- процессор Intel или AMD, с тактовой частотой не менее 1 ГГц;
- ОЗУ объемом 512 Мб;
- НЖМД объемом 20 Гб;
- видеоадаптер с 64 Мб видеопамяти;
- клавиатура;
- манипулятор мышь;
- монитор с разрешением экрана 800 × 600 и частотой обновления 85 Гц.

При проектировании данной системы была использована ЭВМ с техническими характеристиками:

- процессор Intel Core i3 M330 с тактовой частотой 2.3 ГГц;
- ОЗУ объемом 4 Гб;

- НЖМД объемом 500 Гб;
- видеоадаптер NVIDIA GeForce 310M;
- клавиатура;
- манипулятор мышь;
- монитор с разрешением экрана 1366 × 768 и частотой обновления 80 Гц.

## 6 РЕЗУЛЬТАТ РАБОТЫ САПР

### 6.1 Демонстрация результата

Для проведения анализа результата работы системы был поставлен эксперимент. Для данного перечня клиентов был сформирован грузовой автомобиль. Затем он был отправлен в рейс, и маршрут его передвижения был выбран водителем. Маршрут движения автомобиля в процессе доставки был записан при помощи GPS навигатора и проанализирован. После этого на основе исходных данных о количестве клиентов маршрут был сформирован при помощи спроектированной системы. Рассмотрим результат эксперимента.

Необходимо сформировать маршрут для доставки товаров до клиентов. Количество и расположение клиентов в конечном счете не имеет значения. Имеет значение лишь количество и расположение пунктов назначения, в которых они находятся. Количество пунктов назначения и информация о них представлена в таблице 6.1.

После того, как данные были загружены из базы данных, было произведено их преобразование. Данное преобразование заключается в преобразовании абсолютных координат географического положения в экранные. Далее на основе преобразованных данных при помощи подсистемы визуализации была построена карта (рисунок 6.1).

Таблица 6.1 – Перечень населенных пунктов

Название	Широта	Долгота	Индекс
Столовое	52 47.519	41 45.912	1000
Керша	52 52.898	41 52.613	2000
Бондари	52 57	42 04	11000
Бондарский	52 57.11	42 2.3	8000
Б.Шереметьево	53 5.98	42 16.8	7000
Рудовка	53 6	42 23	4000
Осино-Гай	53 1.75	42 24.23	6000
Васильево	53 12	42 14	9000
Свиньино	53 13.2	42 10.7	5000
Пичаево	53 14.1	42 12.1	3000
Тамбов	52 43	41 26	10000
Прибытки	52 54.413	42 3.872	13000
Анненка	53 1.34	42 32.753	12000

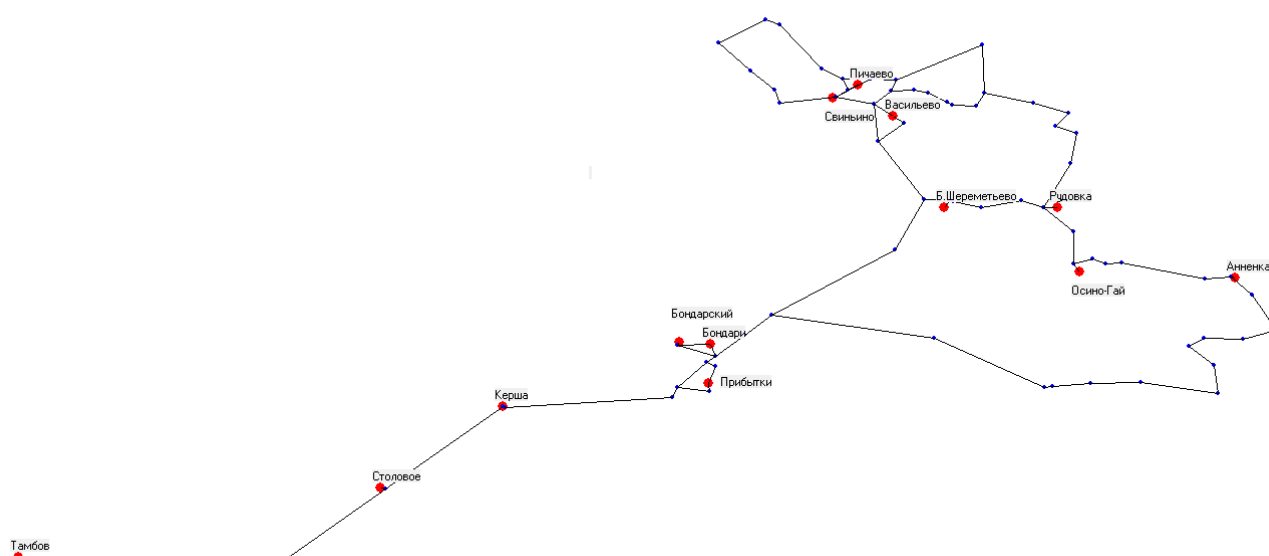


Рисунок 6.1 – Карта заказа

Также в результате преобразования исходных данных и выполнения алгоритма поиска кратчайших вариантов проезда между населенными пунктами была сформирована таблица дорог, фрагмент которой представлен в таблице 6.2.

Таблица 6.2 – Фрагмент таблицы дорог

1000 13 1 14 2000
1000 13 1 14 2 15 18 3 11000
1000 13 1 14 2 15 18 3 19 8000
...
10000 0 13 1 14 2 15 16 13000
10000 0 13 1 14 2 15 18 3 73 20 4 5 44 43 42 41 40 39 38 37 36 35 34 33 12000
13000 17 18 3 73 20 4 5 44 43 42 41 40 39 38 37 36 35 34 33 12000

Первый и последний элементы каждой строки таблицы дорог – это индексы пунктов назначения, а промежуточные значения – это индексы путевых точек. Таблица дорог содержит 78 строк и однозначно определяет порядок проезда между всеми пунктами.

После того, как таблица дорог сформирована, можно приступить к формированию матрицы  $E$ , для того чтобы можно применить метод, при помощи которого выбирается порядок посещения пунктов. Матрица  $E$  – это квадратная матрица, строки и столбцы которой содержат индексы пунктов назначения. Элемент  $E_{i,j}$  матрицы  $E$  содержит значение длины пути из пункта с индексом  $i$  в пункт с индексом  $j$ . В случае, если  $i=j$  или перемещение из пункта  $i$  в  $j$  невозможно,  $E_{i,j}=-1$ . Матрица  $E$ , сформированная на основе таблицы дорог, представлена в таблице 6.3. Значения  $E_{i,j}$  округлены для удобства отображения, однако все расчеты производятся с вещественными числами.

В результате последовательного исключения элементов матрицы, получены значения пар индексов, представленные в таблице 6.4:

Таблица 6.3 – Матрица E

-1	13	68	61	67	68	55	32	63	27	30	78	27
13	-1	56	48	54	56	42	19	51	39	17	65	14
68	56	-1	25	2	33	20	43	6	95	41	43	44
61	48	25	-1	23	9	8	35	20	87	33	19	36
67	54	2	23	-1	31	18	41	4	93	39	41	42
68	56	33	9	31	-1	16	43	28	94	41	12	43
55	42	20	8	18	16	-1	29	14	81	28	26	30
32	19	43	35	41	43	29	-1	38	58	2	52	7
63	51	6	20	4	28	14	38	-1	89	36	37	38
27	39	95	87	93	94	81	58	89	-1	56	104	53
30	17	41	33	39	41	28	2	36	56	-1	51	5
78	65	43	19	41	12	26	52	37	104	51	-1	53
27	14	44	36	42	43	30	7	38	53	5	53	-1

Таблица 6.4 – Пары индексов

Индекс 1	Индекс 2
1000	10000
2000	1000
6000	12000
4000	6000
12000	7000
10000	13000
7000	9000
3000	5000
9000	3000
11000	8000
8000	2000
5000	11000
13000	4000



Это значит, что общая длина маршрута будет минимальна в том случае, если например движение будет осуществляться из пункта с индексом 10000 в пункт с индексом 1000 и т.д. После произведения преобразований полученного результата можно получить порядок перемещения между пунктами.

На рисунке 6.2 представлен результат применения метода, а на рисунке 6.3 представлены таблицы, состоящие из строк, имеющих следующую структуру.

1) Первое поле в каждой строке – это название населенного пункта. Причем строки расположены в порядке посещения пунктов назначения.

2) Все последующие клетки – это индексы путевых точек, которые нужно последовательно посетить для того, чтобы попасть в следующий пункт назначения. Движение из точки, индекс которой представлен в предпоследней клетке предпоследней строки, осуществляется в пункт назначения, представленный в первой строке.

3) В последней ячейке последней строке содержится информация о суммарной длине маршрута.

Столовое	13	0											
Тамбов	0	13	1	14	2	15	16						
Прибытки	17	18	3	73	20	4	5	44	43	42	41	40	
Рудовка	40	39	38										
Осино-Гай	38	37	36	35	34	33							
Анненка	33	34	35	36	37	38	39	40	41	42	43	44	
Б.Шереметь	44	5	11	55									
Васильево	6	7											
Пичаево	7												
Свиньино	7	6	11	5	4	20	73	3					
Бондари	19												
Бондарский	19	3	18	15	2	14							
Керша	14	1	13										
Общий путь								233.28503303356	км				

Рисунок 6.2 – Результат работы системы



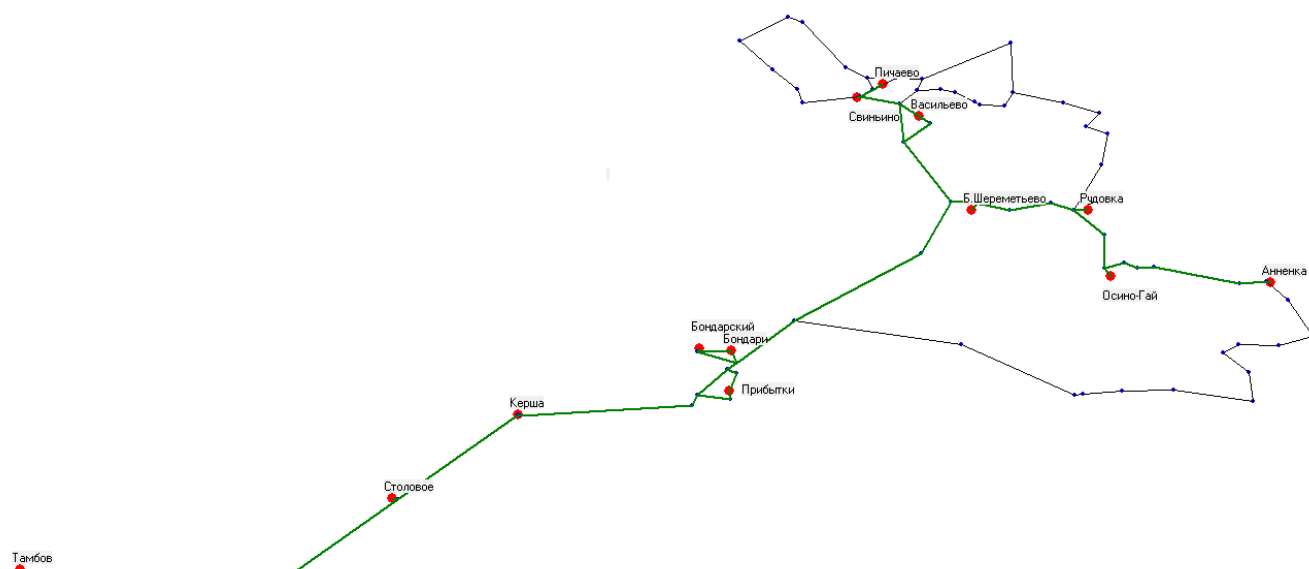


Рисунок 6.5 – Маршрут, вычисленный при помощи системы

## 6.2 Анализ полученного результата

Для полученных индексов путевых точек и названий пунктов назначения можно получить их координаты из базы данных, а затем внести полученную информацию в GPS навигатор для составления маршрута движения или отметить точки на географической карте.

На основе полученных результатов можно сделать вывод о том, что длина маршрута движения, рассчитанного при помощи спроектированной системы короче на величину:

$$dS=254.87 - 233.28=21.59, \text{км}$$

Длина маршрута, рассчитанного с помощью спроектированной системы была бы на 8.5% чем длина маршрута, избранного водителем.

Таким образом, поставленная задача была решена с точки зрения выбранного критерия. Гарантированно маршрут с наименьшей длиной был бы рассчитан при помощи метода полного перебора, но я отказался от его применения, так как для данного количества пунктов назначения (13 штук) количество всех возможных перестановок  $13!$ , а время расчета маршрута составляет около 3 ч, что является недопустимым с точки зрения поставленной задачи, так как маршруты движения формируются в начале рабочего дня и для того, чтобы загрузить товары в машину

необходимо знать последовательность из загрузки в кузов, а следовательно и маршрут движения. Расчет маршрута при помощи спроектированной системы длится 2 минуты.

## ЗАКЛЮЧЕНИЕ

В процессе разработки автоматизированной системы планирования оптимальных грузовых перевозок были проведены следующие работы:

- проведен анализ предметной области;
- разработана структурная схема автоматизированной системы;
- разработана схема работы автоматизированной системы;
- разработана программа, реализующая расчет оптимального маршрута передвижения грузового автомобиля;

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Мирожин Л.Б., Касенов А.Г. Логистика: обслуживание потребителей. – М.: ИНФРА, 2002. – 190 с.
2. Романовский И.В. Алгоритмы решения экстремальных задач. – М.: Наука, 1977. – 352 с.
3. Мудров В.И. Задача о коммивояжере. — М.: «Знание», 1969. — С. 62.
4. Бобровский С. Delphi7. Учебный курс. – СПб.: ПИТЕР, 2008. – 735 с.

ПРИЛОЖЕНИЕ А  
(обязательное)  
СТРУКТУРНАЯ СХЕМА САПР

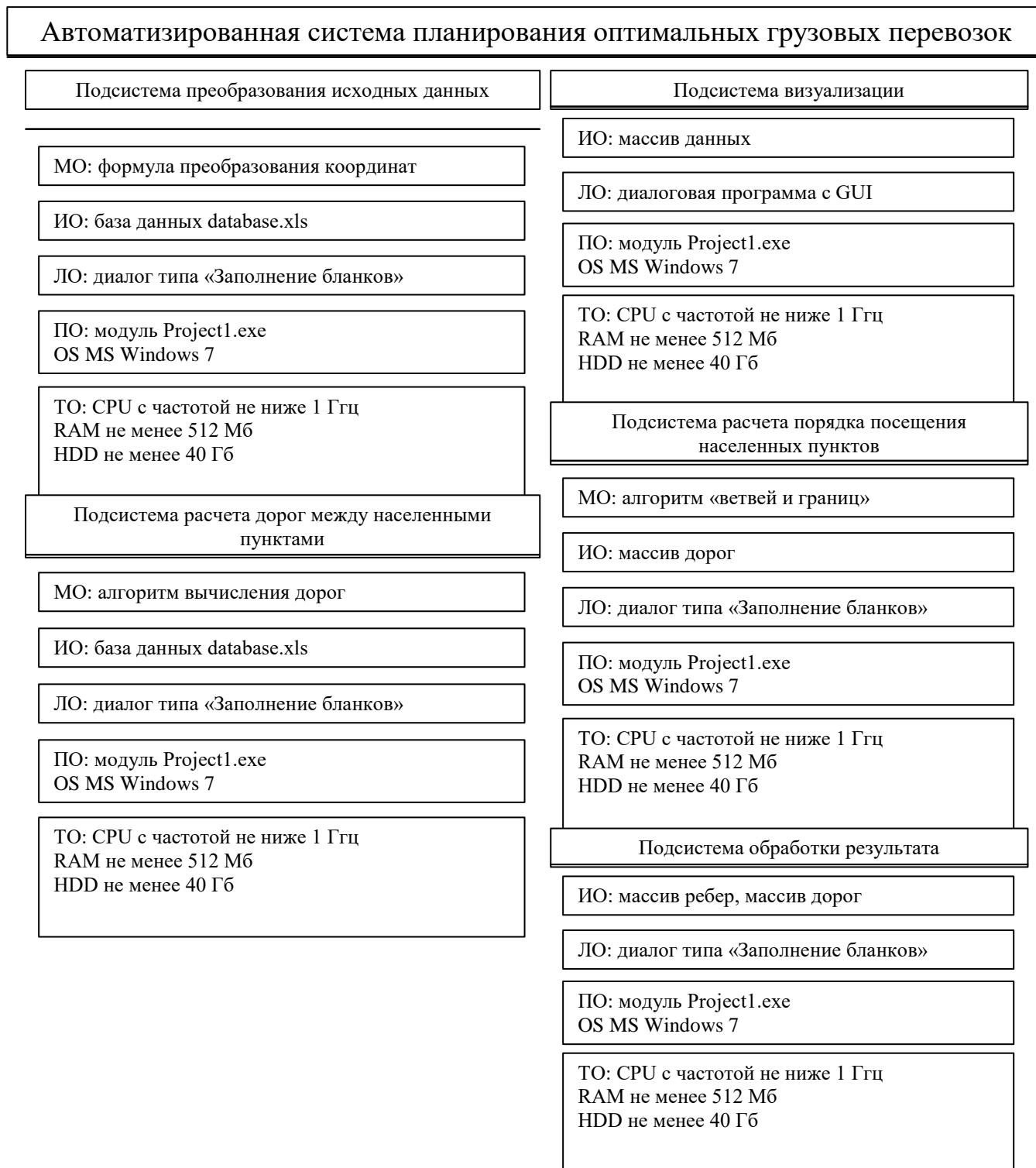


Рисунок А.1 – Структурная схема автоматизированной системы

ПРИЛОЖЕНИЕ Б  
(обязательное)  
СХЕМА РАБОТЫ САПР

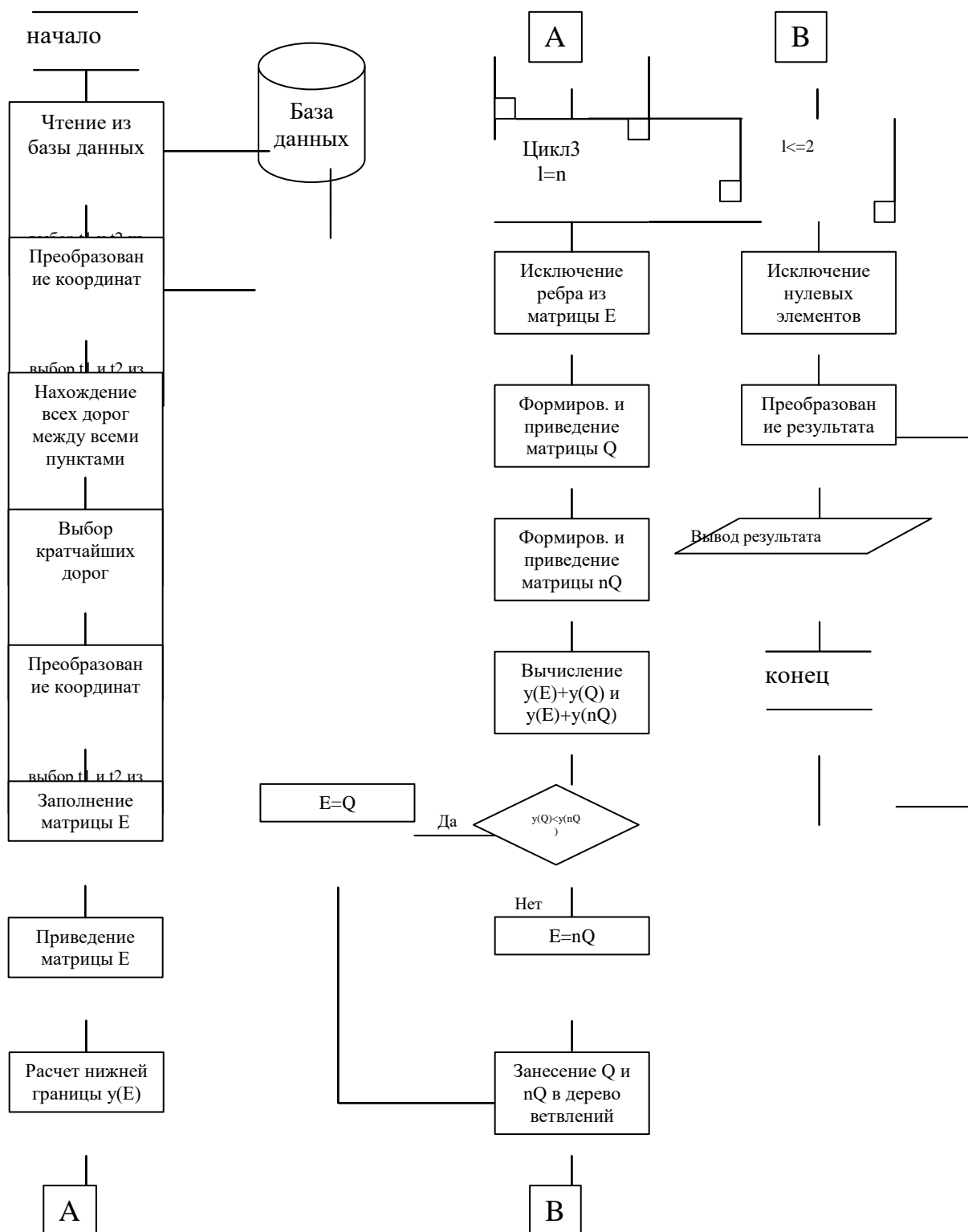


Рисунок Б.1 –Схема работы автоматизированной системы

ПРИЛОЖЕНИЕ В  
(обязательное)  
ДАТАЛОГИЧЕСКАЯ МОДЕЛЬ САПР

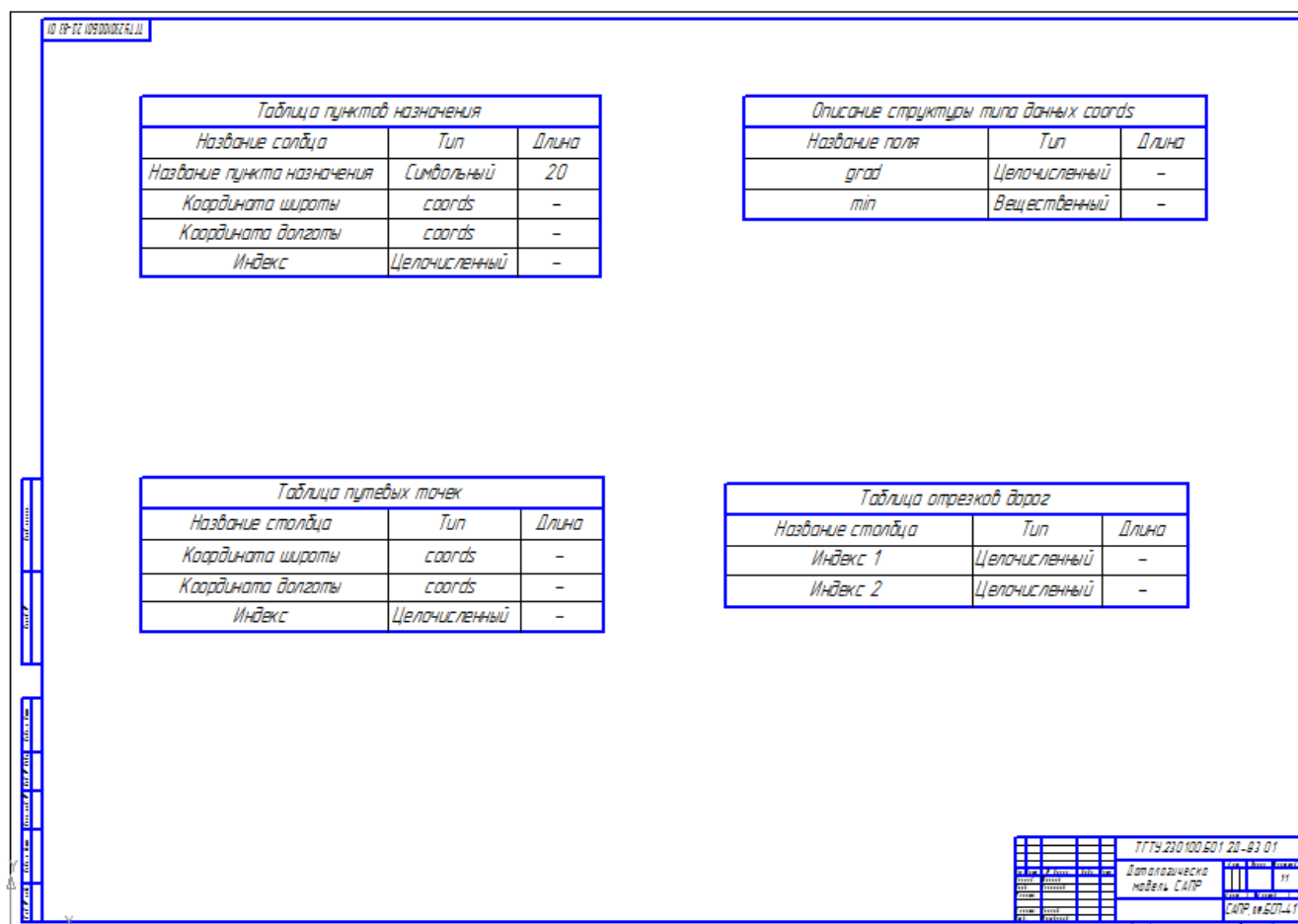


Рисунок В.1 – Даталогическая модель автоматизированной системы

ПРИЛОЖЕНИЕ Г  
(обязательное)  
СХЕМА АЛГОРИТМА

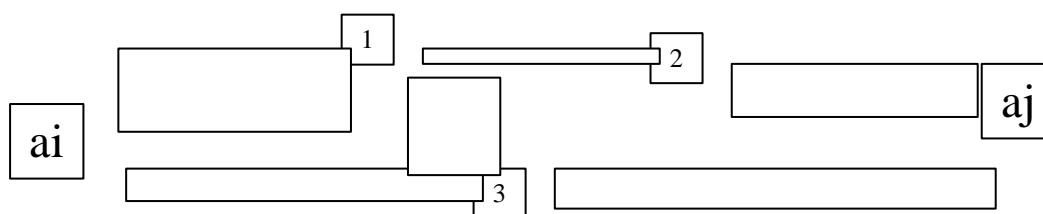


Рисунок Г.1 – Пример расположения населенных пунктов и путевых точек

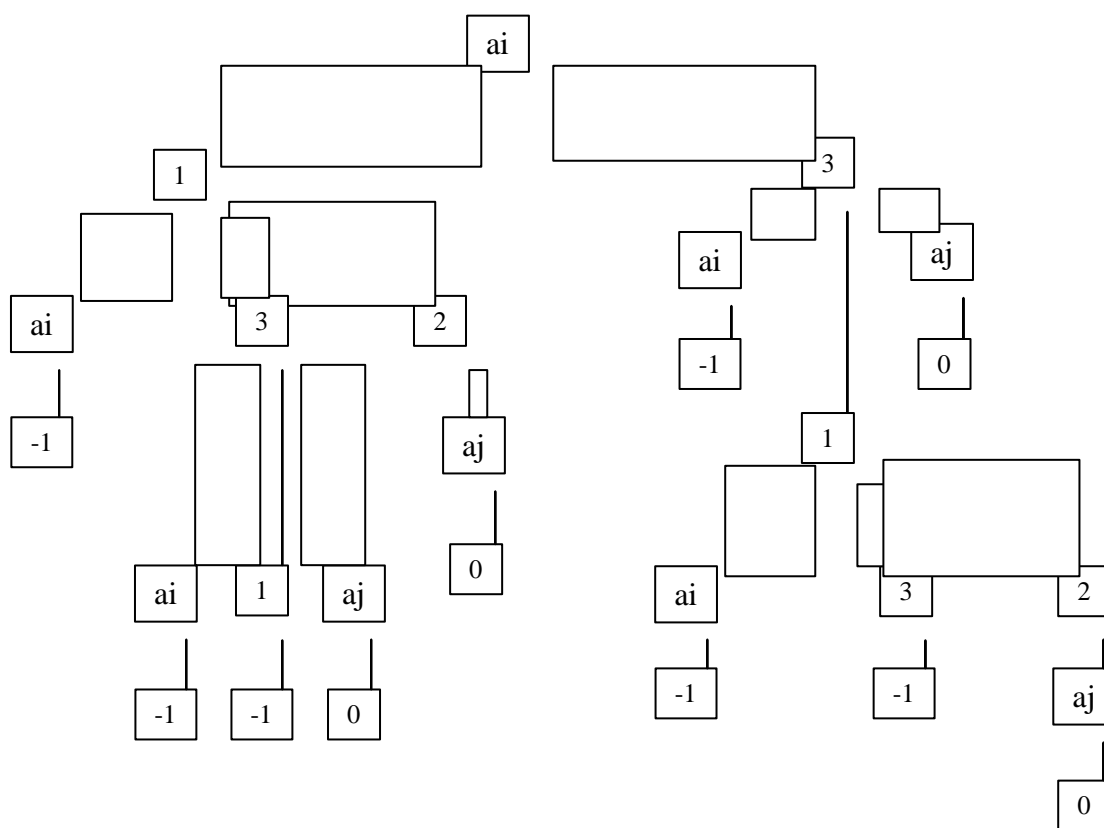


Рисунок Г.2 – Дерево, построенное на основе рисунка Г.1



ПРИЛОЖЕНИЕ Д  
(обязательное)  
РЕЗУЛЬТАТ РАБОТЫ САПР

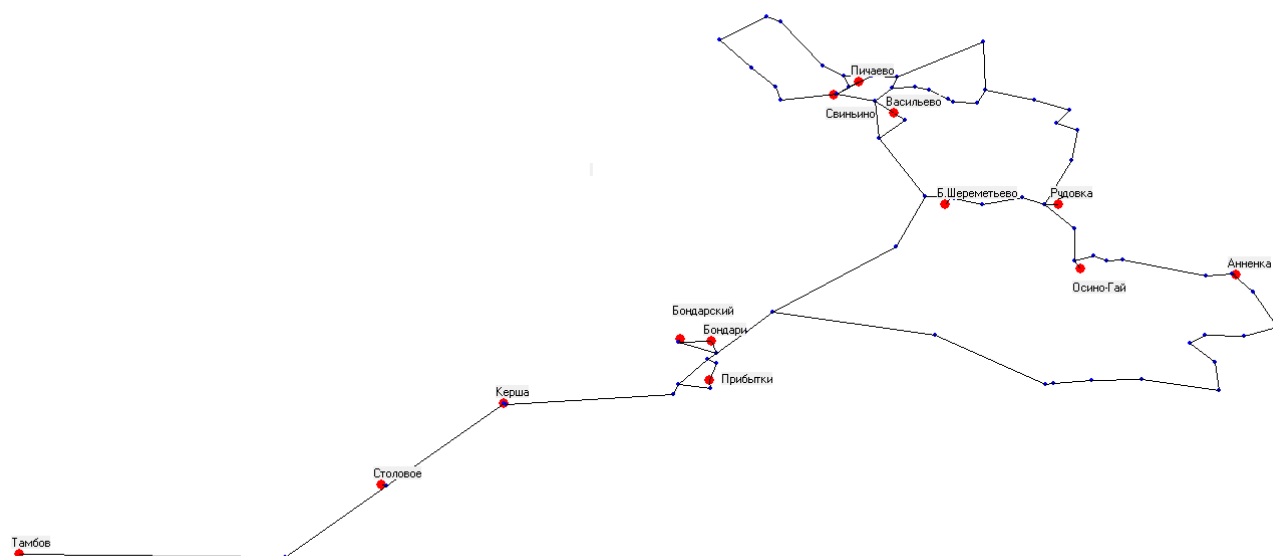


Рисунок Д.1 – Расположение населенных пунктов

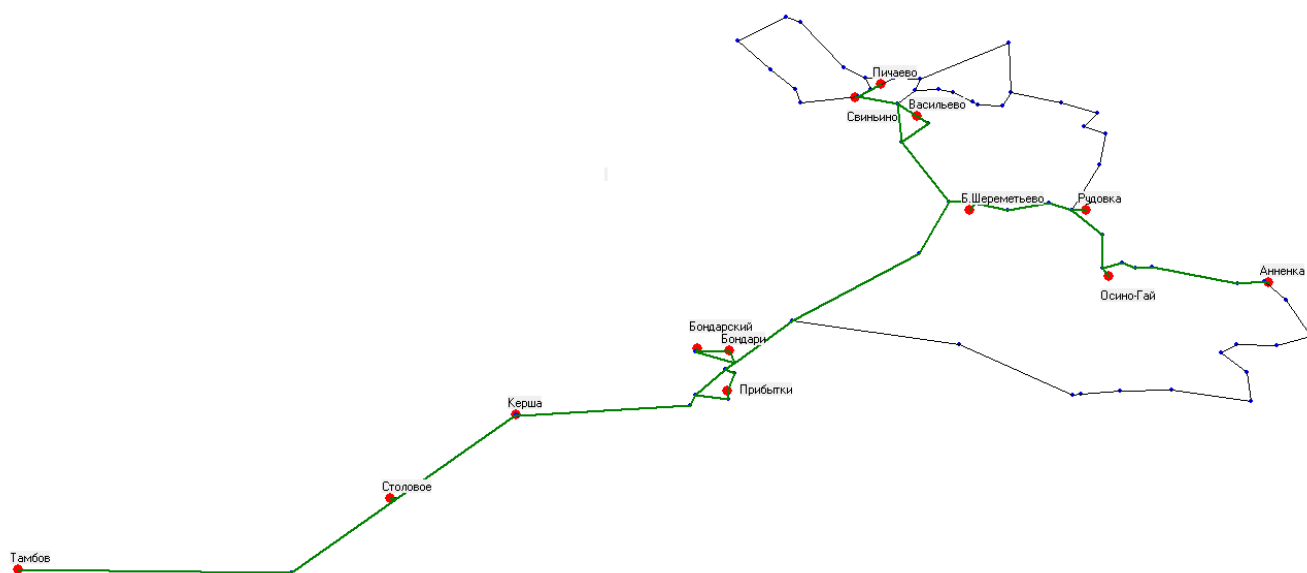


Рисунок Д.2 – Маршрут, рассчитанный при помощи системы